Naval Research Laboratory

Stennis Space Center, MS 39529-5004

NRL/MR/7320--09-9175

User's Manual for the Global Ocean Forecast System (GOFS) Version 3.0

E.J. METZGER

Ocean Dynamics and Prediction Branch
Oceanography Division

O.M. SMEDSTAD
S.N. CARROLL

QinetiQ North America
Planning Systems, Inc.
Slidell, LA

March 12, 2009

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
12-03-2009	Memorandum Report	

4. TITLE AND SUBTITLE

User's Manual for the Global Ocean Forecast System (GOFS) Version 3.0

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
0603207N

6. AUTHOR(S)

E.J. Metzger, O.M. Smedstad,* and S.N. Carroll*

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
73-5094-19-5

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Oceanography Division
Stennis Space Center, MS 39529-5004

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/7320--09-9175

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Space & Naval Warfare Systems Command
2451 Crystal Drive
Arlington, VA 22245-5200

10. SPONSOR / MONITOR'S ACRONYM(S)

SPAWAR

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

*QinetiQ North America, Planning Systems, Inc., Slidell, LA

14. ABSTRACT

This User's Manual describes the sequence of scripts for the Global Ocean Forecast System (GOFS) Version 3.0 that consists of the 1/12° global HYbrid Coordinate Ocean Model (HYCOM) and uses the Navy Coupled Ocean Data Assimilation scheme that is run daily in the normal processing stream at the Naval Oceanographic Office (NAVOCEANO). It is geared toward NAVOCEANO operators rather than a scientific researcher using HYCOM. The appendices contain the individual scripts used in the daily runstream and a troubleshooting guide is included to provide workarounds to some common problems encountered when running the model.

15. SUBJECT TERMS
Numerical ocean modeling
HYCOM

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			E. Joseph Metzger
Unclassified	Unclassified	Unclassified	UL	74	**19b. TELEPHONE NUMBER** *(include area code)* (228) 688-4762

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

TABLE OF CONTENTS

TABLE OF CONTENTS ..... **iii**

TABLE OF FIGURES AND TABLES ..... **iv**

1.0 SCOPE **1**

1.1 INTRODUCTION 1

2.0 REFERENCED DOCUMENTS **2**

2.1 GOVERNMENT AND TECHNICAL REFERENCES 2

3.0 APPLICATION **4**

3.1 DESCRIPTION OF GOFS USAGE 4
3.2 MEMORY AND PROCESSOR ALLOCATION 4
3.3 GOFS V3.0 RUNSTREAM SCRIPTS 4
3.3.1 Main Driver Script *5*
3.3.2 Atmospheric Forcing Generation Script *5*
3.3.3 NCODA Z-level Extraction Script *6*
3.3.4 NCODA Subregion Run Script *6*
3.3.5 NCODA Z-level to HYCOM Vertical Coordinate Conversion Script *6*
3.3.6 HYCOM Run Script *6*

4.0 TROUBLESHOOTING **8**

4.1 THE SYSTEM HAS STOPPED RUNNING 8
4.2 GOFS RUNS FINE FOR A WHILE AND THEN SUDDENLY NEGATIVE LAYER THICKNESSES APPEAR 8
4.3 THE NCODA ANALYSIS OR THE MODEL LOG FILES CONTAINS AN ERROR 9
4.4 CHOLESKY DECOMPOSITION FAILED 10

5.0 GOFS V3.0 OPERATIONAL FLOWCHART **11**

6.0 FUNCTIONAL DESCRIPTION **13**

7.0 NOTES **13**

7.1 ACRONYMS AND ABBREVIATIONS 13

APPENDIX A **14**

SCRIPT 727SHELLSUB_SUBMIT_NCODA.CSH 14

APPENDIX B **19**

SCRIPT 727LSF_WIND_PREP_${IDTGTOD}18_${IDTG}18.COM 19

APPENDIX C **23**

SCRIPT 727EXTR_${SUBREG}_${IDTGTOD}18_${IDTG}18.COM 23

APPENDIX D **30**

SCRIPT 727NOCDA_${SUBREG}_${IDTGTOD}18_${IDTG}18.COM 30

APPENDIX E **44**

SCRIPT 727NCODA2ARCH_${IDTGTOD}18_${IDTG}18.COM 44

APPENDIX F **67**

SCRIPT 727PBS_NCODA_ ${IDTGTOD}18_${IDTG}18.COM 67

TABLE OF FIGURES AND TABLES

FIGURE 1: THE HYCOM/NCODA RUNSTREAM.. 2
TABLE 1: GOFS V3.0 .CSH AND .COM SCRIPTS. 5
TABLE 2: GOFS V3.0 TROUBLESHOOTING GUIDE. 8
TABLE 3: BAROCLINIC TIME STEP AND INCREMENT STEP OPTIONS FOUND IN BLKDAT.INPUT. 9
FIGURE 2: FLOWCHART OF THE GOFS V3.0 RUNSTREAM. 12

1.0 SCOPE

1.1 Introduction

The Global Ocean Forecast System **(**GOFS) Version 3.0 (V3.0) consists of the 1/12° global HYbrid Coordinate Ocean Model (HYCOM) and uses the Navy Coupled Ocean Data Assimilation (NCODA). This User's Manual (UM) describes the sequence of scripts run daily in the normal processing stream at the Naval Oceanographic Office (NAVOCEANO) and is geared toward those users. A more generalized HYCOM User's Manual for running simulations in research mode may be found on the HYCOM consortium website (http://hycom.rsmas.miami.edu). Accompanying documents to this manual are the Software Design Document (SDD) (Wallcraft et al., 2009) and the Phase I Validation Test Report (VTR) (Metzger et al., 2008). It should be noted that when the VTR was written, the system as a whole was referred to as Global Ocean Prediction System (GOPS) V3.0. However, NRL-developed global ocean nowcast/forecast systems have since been renamed the Global Ocean Forecast System (GOFS). It is referred to as such in the SDD and this UM. However, GOPS and GOFS are interchangeable between these documents.

The GOFS V3.0 runstream is depicted in Figure 1. Relative to the nowcast time, the first NCODA ocean analysis is performed at τ = -126 hours with the analysis window for altimeter (and all other) data spanning ±36 (±12) hours. (The first hindcast goes back 5+ days from the nowcast due to late-arriving satellite altimeter data. An examination of the timeliness of the historical altimeter data determined an additional data gain of 18% between four and five days; orbits also improve with the age of the data.).

Manuscript approved February 4, 2009.

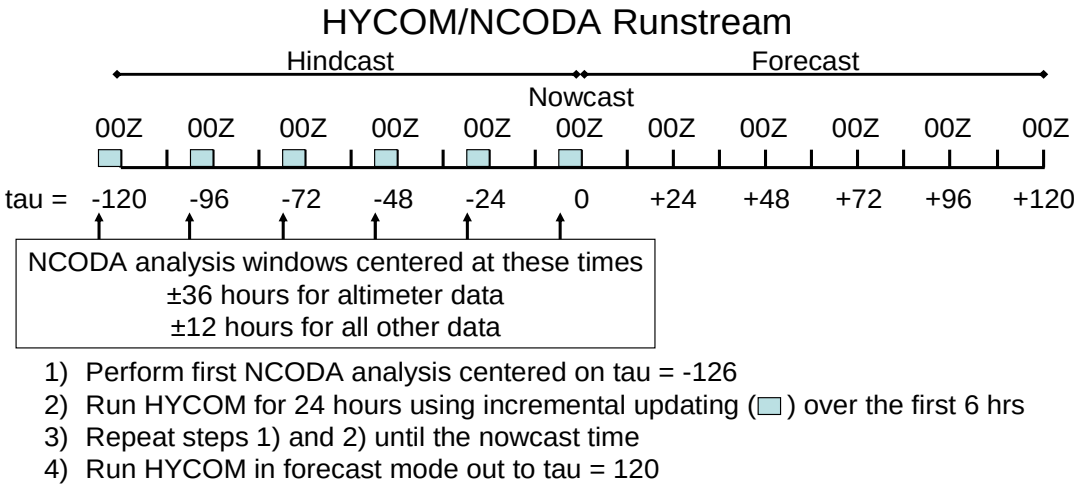


Figure 1: The HYCOM/NCODA runstream. Approximate run times using 379 IBM Power 5+ processors: a) six NCODA analyses – 1.1 hours/analysis = 6.6 hours, b) five HYCOM hindcast days using a 240 second timestep – 0.8 hours/model day = 4.0 hours, c) five HYCOM forecast days using a 240 second timestep – 0.8 hours/model day = 4.0 hours, for a total of d) 14.6 wall-time hours.

After the initial NCODA analysis, HYCOM is run for 24 model hours, with the NCODA analysis incrementally updating the ocean model over the first six hours. Therefore, at 00Z, HYCOM has fully ingested all observational data used in the analysis. The NCODA analysis and HYCOM hindcast cycle repeat themselves daily up to the nowcast time. HYCOM then continues to run in forecast mode out to 120 hours (5 days). The forecast length of five days was chosen in part because atmospheric forcing typically is available that far out in time. As currently configured, the atmospheric forcing is blended toward climatology after the forecast output ends, so the system could run in forecast mode out to ~30 days. However, this would require considerable additional computational resources.

2.0 REFERENCED DOCUMENTS

2.1 Government and Technical References

Cummings, J. and Carroll, S., (2006). Software User's Manual for the Navy Coupled Ocean Data Assimilation (NCODA) System, *NRL Tech. Rpt.*, MRY-001-06, Naval Research Laboratory, Monterey, CA.

Metzger, E.J., Smedstad, O.M., Thoppil, P., Hurlburt, H.E., Wallcraft, A.J., Franklin, D.S., Shriver, J.F. and Smedstad, L.F., (2008). Validation Test Report for the Global

Ocean Prediction System V3.0 – 1/12° HYCOM/NCODA: Phase I, *NRL Memo. Rpt.*, NRL/MR/7320—08-9148, Ocean Modeling Division, Naval Research Laboratory, Stennis Space Center, MS.

Wallcraft, A.J., Metzger E.J. and Carroll S.N., (2009). Software Design Description for the HYbrid Coordinate Ocean Model (HYCOM) Version 2.2. *NRL Memo. Rpt.*, NRL/MR/7320—09-9166, Ocean Modeling Division, Naval Research Laboratory, Stennis Space Center, MS.

3.0 APPLICATION

3.1 Description of GOFS Usage

This manual describes the procedures for running the Global Ocean Forecast System Version 3.0. It is designed for NAVOCEANO personnel who will be running GOFS in their operational queues. The user may also refer as needed to the HYCOM Software Design Description (SDD), which describes the code, physics and basic equations (Wallcraft et al., 2009), and the Validation Test Report (Metzger et al., 2008) for this system. Additionally, the NCODA UM (Cummings and Carroll, 2006) may be of use.

3.2 Memory and Processor Allocation

HYCOM can run on any number of processors and its footprint can be changed accordingly. It is currently configured to use 619 processors, but 596 or 713 processors are other multiples for which the code is compiled. Memory requirements for NCODA vary depending upon the amount of data that goes into the analysis.

3.3 GOFS V3.0 Runstream Scripts

For a daily integration of GOFS V3.0, a sequence of scripts are generated and run. All scripts contain environmental variables ${E} and ${EXPT} which define the experiment number. The example that follows uses

```
setenv E       727
setenv EXPT    expt_72.7.
```

If a new experiment number is desired, these variables would have to be changed in all scripts and the appropriate directory structures would need to be created. Figure 2 is a schematic flowchart of the system and all scripts are provided in Appendices A-F.

Several scripts are used to direct the running of GOFS V3.0. These are summarized in Table 1 below. The script names often begin with the experiment number, in this example, 727.

Table 1: GOFS V3.0 .csh and .com scripts.

Script	Description
727shellsub_submit_ncoda.csh	Main driver script. Generates scripts for each daily run. See Section 3.3.1 and Appendix A.
727lsf_wind_prep_${idtgtod}18_${idtg}18.com	Atmospheric forcing generation script. See Section 3.3.2 and Appendix B.
727extr_${subreg}_${idtgtod}18_${idtg}18.csh	Extracts NCODA *z*-levels and surface fields from the HYCOM archive file for each NCODA subregion. See Section 3.3.3 and Appendix C.
727ncoda_${subreg}_${idtgtod}18_${idtg}18.csh	Runs the NCODA analysis for each subregion. See Section 3.3.4 and Appendix D.
727ncoda2arch_${idtgtod}18_${idtg}18.csh	Converts the NCODA analysis on *z*-levels to HYCOM vertical coordinates. See Section 3.3.5 and Appendix E.
727lsf_ncoda_${idtgtod}18_${idtg}18.com	Runs HYCOM. See Section 3.3.6 and Appendix F.

3.3.1 Main Driver Script

The main driver script may be found at /u/home/ooc/models/hycom/GLBa0.08/expt_72.7/mvoi/727shellsub_submit_ncoda.csh. Running as a cron job, it generates the set of scripts needed for each daily run. Most of the work is simply script creation, but its final task is submitting a job to create the atmospheric forcing needed by GOFS V3.0. A subsequent sequence of scripts then gets executed. Appendix A provides the complete script for viewing.

The following command sets the number of hindcast days. It is currently set to five days.

```
#
# --- Go back 5 days from today to do a 5 day hindcast
# --- idtg: date-time-group for today minus 5 days
#
set idtg=`/u/home/${user}/bin/addndays yyyymmdd ${idtgtod} -5`
#
```

3.3.2 Atmospheric Forcing Generation Script

The atmospheric forcing generation script is found at /scr/ooc/data/hycom/GLBa0.08/expt_72.7/logs/727lsf_wind_prep_${idtgtod}18_${idtg}18.com

Here environmental variable ${idtgtod} is the date-time-group for today (e.g. 20081206) and variable ${idtg} is the date-time-group for today minus the hindcast length (e.g. 20081201). See Appendix B for a copy of the entire script.

3.3.3 NCODA Z-level Extraction Script

This script extracts NCODA z-levels from the HYCOM archive file.
/scr/ooc/data/hycom/GLBa0.08/expt_72.7/logs/72.7extr_${subreg}_${idtgtod}18_${idtg}18.csh
Environmental variable ${subreg} is the subregion identifier. Currently there are seven: ANTarc, ARCatl, ARCocn, ARCpac, MERat1, MERin1, and MERpa1. Appendix C provides the complete script and an example from MERpa1.

3.3.4 NCODA Subregion Run Script

This script runs the NCODA analysis for each subregion. /scr/ooc/data/hycom/GLBa0.08/expt_72.7/logs/727ncoda_${subreg}_${idtgtod}18_${idtg}18.csh. All subregions are run in parallel. These jobs also start the NCODA plotting and file transfer to the permanent storage machine. See Appendix D for the complete script and an example from MERpa1.

3.3.5 NCODA Z-level to HYCOM Vertical Coordinate Conversion Script

This script converts the NCODA analysis on z-levels to HYCOM vertical coordinates. /scr/ooc/data/hycom/GLBa0.08/expt_72.7/logs/727ncoda2arch_${idtgtod}18_${idtg}18.csh. This job is submitted by one of the NCODA analysis scripts and will wait until all the subregions are completed before it executes. Upon completion, it submits a plotting job to HYCOM. Appendix E shows a copy of this script.

3.3.6 HYCOM Run Script

/scr/ooc/data/hycom/GLBa0.08/expt_72.7/logs/727lsf_ncoda_${idtgtod}18_${idtg}18.com is the script that runs HYCOM. At the end of this script, the next day is submitted by

727shellsub_daily.csh, which is very similar to the cron script that initiates the entire sequence. See Appendix F for the complete run script.

4.0 TROUBLESHOOTING

Table 2 provides a brief troubleshooting guide for when problems are encountered. It is followed by a more detailed discussion of each specific problem.

Table 2: GOFS V3.0 Troubleshooting Guide.

Problem	Solution
1) System has stopped running.	Check log files in /scr/ooc/data/hycom/GLBa0.08/expt_72.7/logs to find the part of the system where the problem occurred.
2) HYCOM stopped running due to negative layer thicknesses, i.e. "neg. dp" is seen in the log file.	**a)** Reduce the baroclinic time step (baclin) in /u/home/ooc/models/hycom/GLBa0.08/expt_72.7/blkdat.input. If unsuccessful, try, **b)** Turn off the incremental updating (in **blkdat.input**) by setting 'incflg' = incremental update flag (0=no, 1=yes, 2=full-velocity).
3) Error message: Cannot load ntbl_windows on all selected nodes...	This is a machine problem and resubmitting the script usually works.
4) Cholesky decomposition failed error in a NCODA subregion.	Adjust the diagonal of the covariance matrices by changing the *oanl* namelist variable err_plus() = 1. Set it to a negative value after the analysis has gotten past the problem, see **oanl.h**, the NCODA namelist file.

4.1 The System Has Stopped Running

Explanation: One of the jobs has died for some reason.

Solution: See /scr/ooc/data/hycom/GLBa0.08/expt_72.7/logs/.

At the system prompt, enter

```
ls –lat | more
```

to see where the run died. Check the appropriate log files for errors.

4.2 GOFS Runs Fine For A While And Then Suddenly Negative Layer Thicknesses Appear

The user may see "neg. dp" in the log file.

Explanation: Model is unstable.

Solution 1: Reduce the baroclinic time step (baclin) in

`/u/home/ooc/models/hycom/GLBa0.08/expt_72.7/blkdat.input`

Use the table in ***blkdat.input***, as illustrated in Table 3 below, to change the baroclinic time step and incstp so that the increments are put in over 6 hours.

Table 3: Baroclinic time step and increment step options found in blkdat.input.

baclin	incstp
75	288
120	180
150	144
240	90
300	72

Solution 2: If lowering the time step does not work, the observations might be bad. Turn the assimilation off for the day. In

`/u/home/ooc/models/hycom/GLBa0.08/expt_72.7/blkdat.input`

change

`2 'incflg'` = incremental update flag (0=no, 1=yes, 2=full-velocity)

to

`0 'incflg'` = incremental update flag (0=no, 1=yes, 2=full-velocity).

It is possible that the user must go back to the previous day, since the instability could have begun earlier.

NOTE: Remember to change *incflg* back to 2 after the problem day has run successfully.

4.3 The NCODA Analysis or The Model Log Files Contains An Error

The error could be, for example,

```
Sat Nov 29 11:32:09 GMT 2008 [poejob]: Cannot load
ntbl_windows on all selected nodes ...
```

Explanation: An error such as this usually indicates a machine problem. Resubmitting the job usually solves the problem.

Solution: Resubmit the script that failed. If it is one of the NCODA analysis scripts, uncomment the line

```
#/u/home/wallcraf/bin/msub
${E}ncoda2arch_${idtgtod}_${curr_dtg}.csh 1
```

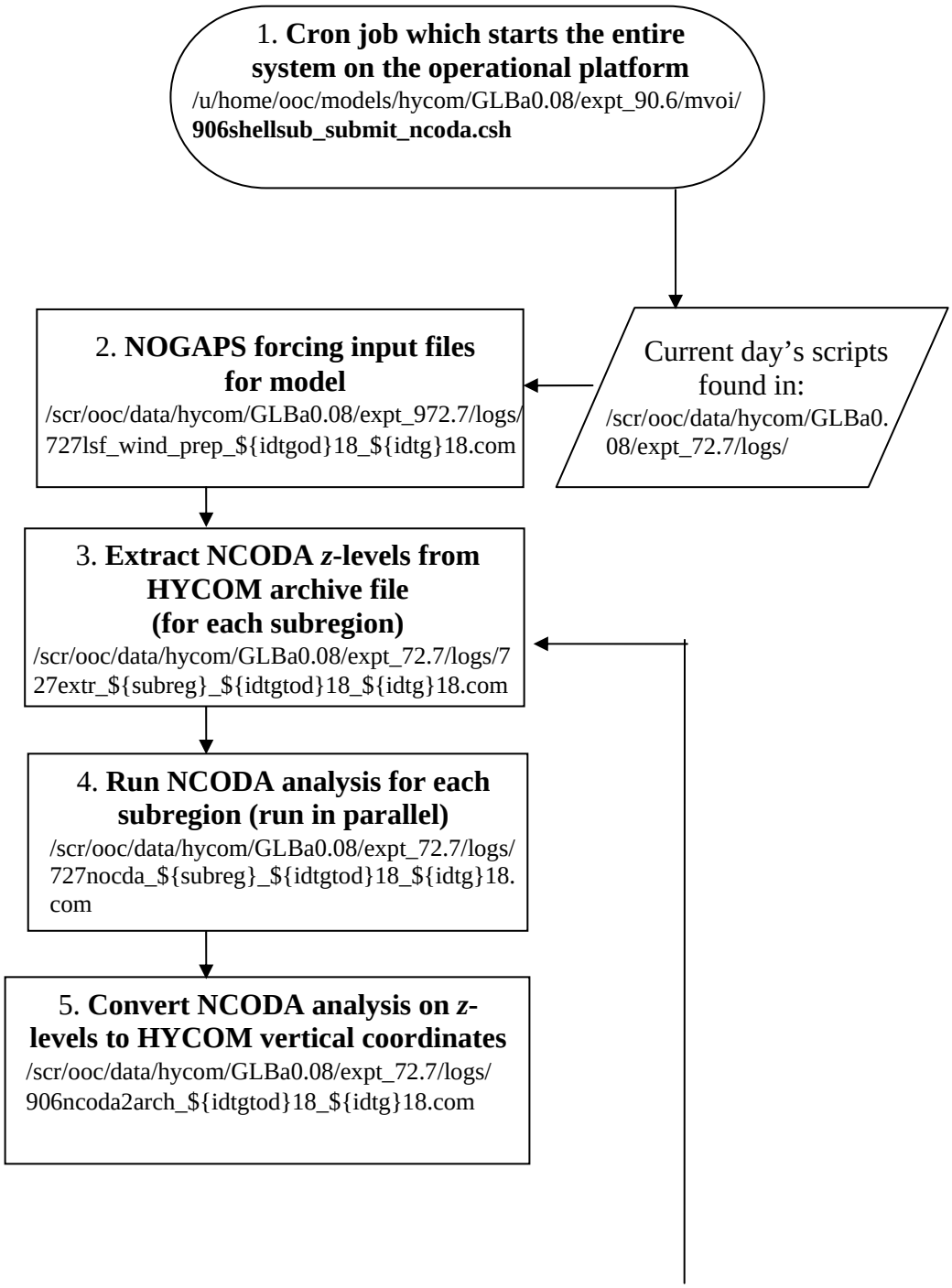
in e.g., ***727ncoda_ARCatl_2008112918_2008112518.csh*** at the bottom of the script. This is done so that the NCODA-to-HYCOM script is submitted and the model run continues. Make sure that the ***ncoda2arch*** script is not already running. If so, kill the job before resubmitting the NCODA analysis for the subregion.

4.4 Cholesky Decomposition Failed

Explanation: Either the background error is too large or the observation error is too small. This results in very small diagonal elements. With converging grid locations and super-obs based on grid i,j indices, the data can be very close and the off-diagonal correlations can be large. The result is a poorly conditioned matrix.

Solution: Adjust the diagonal of the covariance matrices by changing the *oanl* namelist variable *err_plus*() = 1. Set it to a negative value after the analysis has moved past the problem. See the NCODA namelist file ***oanl.h.***

5.0 GOFS V3.0 OPERATIONAL FLOWCHART



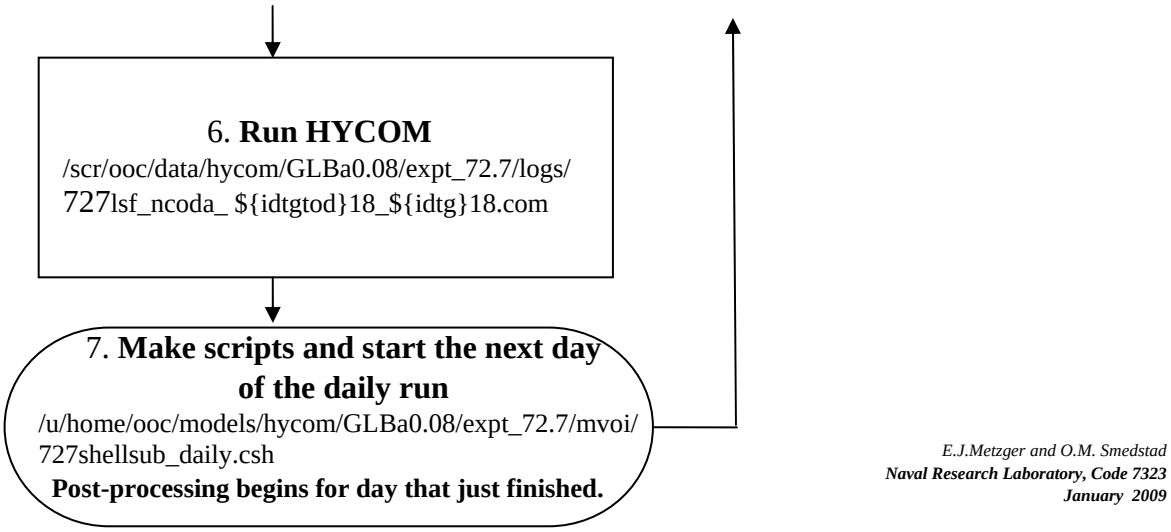


Figure 2: Flowchart of the GOFS V3.0 runstream. All scripts are provided in Appendices A-F.

6.0 FUNCTIONAL DESCRIPTION

For a discussion of the HYCOM 2.2 model theory and code description see the accompanying Software Design Description (Wallcraft *et al.*, 2009).

7.0 NOTES

7.1 Acronyms and Abbreviations

Acronym	Definition
GOFS	Global Ocean Forecast System
HYCOM	HYbrid Coordinate Ocean Model
MPI	Message Passing Interface
NCAR	National Center for Atmospheric Research
NCODA	Navy Coupled Ocean Data Assimilation
NRL	Naval Research Laboratory
SDD	Software Design Description
UM	User's Manual
VTR	Validation Test Report

APPENDIX A

Script 727shellsub_submit_ncoda.csh

This is the main driver script.

```
#! /bin/csh -f
#
# --- This script generates the first set of scripts needed for todays
# --- HYCOM-NCODA run. Most of the work is simply script creation, but
# --- the last thing it does is submit a job to create the forcing needed
# --- by the ocean model. A subsequent sequence of scripts gets executed
# --- in turn.
#
# --- Originally scripted by O.M. Smedstad, QinetIQ-PSI, Inc.
# --- ole.smedstad.ctr@nrlssc.navy.mil
# --- office phone: 228-688-4365
#
set path=( $path /usr/bin/X11 /u/home/wallcraf/bin /u/home/${user}/bin . )
echo $path
#
# --- The environmental variables ${E} and ${EXPT} define the experiment
# --- number. If setting up for a new experiment, these need to be changed
# --- throughout all scripts.
#
setenv E 727
setenv REG1 GLBa0.08
setenv EXPT expt_72.7
setenv MVOI /u/home/${user}/hycom/${REG1}/${EXPT}/mvoi
setenv LOGS /scr/${user}/hycom/${REG1}/${EXPT}/logs
#
# --- Make directories in case they do not exist
#
mkdir -p ${LOGS}
mkdir -p /scr/${user}/hycom/${REG1}/${EXPT}/data/wind
mkdir -p /scr/${user}/hycom/${REG1}/${EXPT}/data/flux
mkdir -p /scr/${user}/hycom/${REG1}/${EXPT}/data/ssta
mkdir -p /scr/${user}/hycom/${REG1}/${EXPT}/data/pcip
mkdir -p /scr/${user}/hycom/${REG1}/${EXPT}/data/wspd
mkdir -p /scr/${user}/hycom/${REG1}/${EXPT}/data/incup
#
# --- The time of the NCODA analysis
#
setenv HR  18
setenv HR2 ${HR}
#
# --- The day model is submitted (run) (format 20070101)
# --- idtgtod: date-time-group for today
#
setenv idtgtod `date +%Y%m%d`
#setenv idtgtod $1
#
# --- Go back 5 days from today to do a 5 day hindcast
# --- idtg: date-time-group for today minus 5 days
#
set idtg=`/u/home/${user}/bin/addndays yyyymmdd ${idtgtod} -5`
#
```

```
# --- Total number of days to create forcing files
#
#set maxnumdays=15
set maxnumdays=3
#
# --- Number of days of model integration between each NCODA analysis
#
setenv nmdays 1
#
# --- idtgm1: one day prior to the system start date
# --- idtgtodm1: one day prior to the system start date
#
setenv idtgm1 `/u/home/${user}/bin/addndays yyyymmdd ${idtg} -1`
setenv idtgtodm1 `/u/home/${user}/bin/addndays yyyymmdd ${idtgtod} -1`
#
echo 'idtgtod=' ${idtgtod} ' idtg=' ${idtg}
#
# --- Start job at this time
#
setenv month `echo ${idtgtod} |cut -c5-6`
setenv day   `echo ${idtgtod} |cut -c7-8`
setenv nxtd 20:00
#
# --- Delete forcing files that may exist with idtg
#
/bin/rm -f /scr/${user}/hycom/${REG1}/${EXPT}/data/ssta/*${idtg}*
/bin/rm -f /scr/${user}/hycom/${REG1}/${EXPT}/data/wind/*${idtg}*
/bin/rm -f /scr/${user}/hycom/${REG1}/${EXPT}/data/flux/*${idtg}*
/bin/rm -f /scr/${user}/hycom/${REG1}/${EXPT}/data/wspd/*${idtg}*
/bin/rm -f /scr/${user}/hycom/${REG1}/${EXPT}/data/pcip/*${idtg}*
#
cd /u/home/${user}/hycom/${REG1}/${EXPT}
#
# --- Make necessary scripts for each NCODA analysis subregion
# --- for the given date
#
foreach REG( ANTarc MERat1 MERin1 MERpa1 ARCatl ARCpac ARCocn )
#
# --- Generate the script to plot NCODA fields
#
  /bin/rm ${LOGS}/${E}plot_${REG}_${idtgtod}${HR}_${idtg}${HR}*.{csh,log}
  awk -f ${MVOI}/MVOI.awk tod=${idtgtod} hr=${HR} t1=${idtg} tr=${REG} \
   ${MVOI}/${E}plot.csh >                                              \
   ${LOGS}/${E}plot_${REG}_${idtgtod}${HR}_${idtg}${HR}.csh
#
# --- Generate the script to rcp NCODA plots
#
  /bin/rm ${LOGS}/${E}plotrcp_${REG}_${idtgtod}${HR}_${idtg}${HR}*.{csh,log}
  awk -f ${MVOI}/MVOI.awk hr=${HR} tod=${idtgtod} t1=${idtg} tr=${REG} \
  ${MVOI}/${E}plotrcp.csh >                                           \
  ${LOGS}/${E}plotrcp_${REG}_${idtgtod}${HR}_${idtg}${HR}.csh
#
# --- Generate the script to perform the NCODA analysis
#
  /bin/rm ${LOGS}/${E}ncoda_${REG}_${idtgtod}${HR}_${idtg}${HR}*.{csh,log}
  awk -f ${MVOI}/MVOI.awk reg=${REG} tod=${idtgtod}${HR} t1=${idtg}${HR} \
   t2=1 t3=${idtgm1}${HR} t4=${E}ncoda_${REG}_${idtg}.csh               \
   ${MVOI}/${E}ncoda_${REG}.job >                                       \
```

```
   ${LOGS}/${E}ncoda_${REG}_${idtgtod}${HR}_${idtg}${HR}.csh
#
# --- Generate the script to extract HYCOM fields for NCODA analysis
# --- todm1=idtgtodm1 for the first analysis
#
  /bin/rm ${LOGS}/${E}extr_${REG}_${idtgtod}${HR}_${idtg}${HR}*.{csh,log}
  awk -f ${MVOI}/MVOI_daily.awk nxtd=${nxtd} calnmb=1 hr=${HR} hr2=${HR}
t1=${idtg} \
   tod=${idtgtod} tod1=${idtgtod} todm1=${idtgtodm1}
${MVOI}/${E}extr_${REG}.csh_zi >        \
   ${LOGS}/${E}extr_${REG}_${idtgtod}${HR}_${idtg}${HR}.csh
#
# --- Change file permissions
#
  chmod u+rx ${LOGS}/*${idtg}*
#
end
#
# --- Generate script to interpolate to uniform 1/12 deg grid (GLBu0.083)
# --- and write netcdf files
#
#/bin/rm ${LOGS}/${E}subreg_GLBu0.083_${idtgtod}${HR}_${idtg}${HR}*.{csh,log}
#awk -f ${MVOI}/MVOI.awk hr=${HR} tod=${idtgtod}00 t1=${idtgtod}00      \
#t2=1 t3=${idtgm1}00 ${MVOI}/../../subregion/${E}subreg_GLBu0.083.com >  \
# ${LOGS}/${E}subreg_GLBu0.083_${idtgtod}${HR}_${idtg}${HR}.csh
#
# --- Generate the NCODA prep script (get all data for the period idtg to
# --- idtgtod) not necessary for a real time job if the data are already on
# --- the machine
#
/bin/rm ${LOGS}/${E}ncoda_PREP_${idtgtod}${HR}_${idtg}${HR}*.{csh,log}
awk -f ${MVOI}/MVOI.awk nxtd=${nxtd} reg=PREP1 hr=${HR} tod=${idtgtod}00 \
 t1=${idtgtod}00 t2=1 t3=${idtgm1}00 t4=${E}ncoda_${REG1}_${idtg}.csh    \
 ${MVOI}/${E}ncoda_PREP.job >                                           \
 ${LOGS}/${E}ncoda_PREP_${idtgtod}${HR}_${idtg}${HR}.csh
#
# --- Generate the script to convert a HYCOM restart file to
# --- a HYCOM archive file
#
#/bin/rm ${LOGS}/${E}rest2archv_${idtgtod}${HR}_${idtg}${HR}.csh
#awk -f ${MVOI}/MVOI.awk nxtd=${nxtd} hr=${HR} tod=${idtgtod} tod1=${idtgtod} \
# t1=${idtg} ${MVOI}/${E}restart_archv_1.csh >                                \
# ${LOGS}/${E}rest2archv_${idtgtod}${HR}_${idtg}${HR}.csh
#
# --- Generate the script to convert an NCODA file to a HYCOM archive file
#
/bin/rm ${LOGS}/${E}ncoda2arch_${idtgtod}${HR}_${idtg}${HR}*.{csh,log}
awk -f ${MVOI}/MVOI.awk hr=${HR} tod=${idtgtod} tod1=${idtgtod} t1=${idtg} \
todm1=${idtgtodm1} ${MVOI}/${E}ncoda2arch_all.csh >
\
 ${LOGS}/${E}ncoda2arch_${idtgtod}${HR}_${idtg}${HR}.csh
#
# --- Generate the script to copy log files to newton
#
/bin/rm ${LOGS}/${E}cplogs_${idtgtod}${HR}_${idtgm1}${HR}*.{csh,log}
awk -f ${MVOI}/MVOI.awk hr=${HR} t1=${idtgm1} tod=${idtgtod} \
 ${MVOI}/../cplogs.csh > ${LOGS}/${E}cplogs_${idtgtod}${HR}_${idtgm1}${HR}.csh
#
```

```
# --- Generate the script to copy gmeta files to NRL anonymous ftp server
#
/bin/rm ${LOGS}/${E}cpgmeta_${idtgtod}${HR}_${idtgm1}${HR}*.{csh,log}
awk -f ${MVOI}/MVOI.awk hr=${HR} t1=${idtgm1} tod=${idtgtod} \
 ${MVOI}/../cpgmeta.csh > ${LOGS}/${E}cpgmeta_${idtgtod}${HR}_${idtgm1}${HR}.csh
#
# --- Generate the script to copy archv_1 files to newton
#
/bin/rm ${LOGS}/${E}cparchv_1_${idtgtod}${HR}_${idtgm1}${HR}*.{csh,log}
awk -f ${MVOI}/MVOI.awk hr=${HR} t1=${idtgm1} tod=${idtgtod} \
 ${MVOI}/../cparchv_1.csh >                                  \
 ${LOGS}/${E}cparchv_1_${idtgtod}${HR}_${idtgm1}${HR}.csh
#
# --- Generate the script to copy files to NRL anonymous ftp server
#
/bin/rm ${LOGS}/ztar_${idtgtod}${HR}_${idtg}${HR}.csh
awk -f ${MVOI}/MVOI.awk hr=${HR} hr=${HR} t1=${idtg} tod=${idtgtod} \
 ${MVOI}/../${E}maktar_glba0.08.com >
${LOGS}/ztar_${idtgtod}${HR}_${idtg}${HR}.csh
#
# --- Generate the script to get the BOGUS files from file server
#
/bin/rm ${LOGS}/getfro_${idtgtod}${HR}.sh
awk -f ${MVOI}/MVOI.awk hr=${HR} hr=${HR} t1=${idtg} tod=${idtgtod} \
 ${MVOI}/../getfro.sh > ${LOGS}/getfro_${idtgtod}${HR}.sh
#
cd ${LOGS}
#
# --- Submit job getting the BOGUS files
#
/u/home/wallcraf/bin/q_navo getfro_${idtgtod}${HR}.sh
#
# --- Generate the script to create forcing files for the number of
# --- days in todays run (maxnumdays)
#
/bin/rm ${LOGS}/${E}cpforcing_${idtgtod}${HR}_${idtg}${HR}*.{sh,log}
awk -f ${MVOI}/../HYCOM.awk nmdays=${maxnumdays} t0=${E}pbs_${idtg} hr=${HR} \
 tod=${idtgtod} t1=${idtg} ${MVOI}/../${E}cpforcingtomsrc_real.sh>      \
 ${LOGS}/${E}cpforcing_${idtgtod}${HR}_${idtg}${HR}.sh
#
/bin/rm ${LOGS}/${E}pbs_wind_prep_${idtgtod}${HR}_${idtg}${HR}*.{com,log}
awk -f ${MVOI}/../HYCOM.awk nmdays=${maxnumdays} t0=${E}pbs_${idtg} hr=${HR} \
 tod=${idtgtod} t1=${idtg} ${MVOI}/../${E}pbs_wind_prep.com >      \
 ${LOGS}/${E}pbs_wind_prep_${idtgtod}${HR}_${idtg}${HR}.com
#
# --- Generate the HYCOM run script for today
#
/bin/rm ${LOGS}/${E}pbs_ncoda_${idtgtod}${HR}_${idtg}${HR}*.{com,log}
awk -f ${MVOI}/../HYCOM.awk nmdays=${nmdays} t0=${E}pbs_${idtg} hr=${HR} \
 hr2=${HR} tod=${idtgtod} t1=${idtg} ${MVOI}/../${E}pbs_ncoda.com >      \
 ${LOGS}/${E}pbs_ncoda_${idtgtod}${HR}_${idtg}${HR}.com
#
# --- Submit the forcing preparation job
#
# on XT5
/u/home/wallcraf/bin/q_navo ${E}cpforcing_${idtgtod}${HR}_${idtg}${HR}.sh
# on IBM
#/u/home/wallcraf/bin/q_navo ${E}pbs_wind_prep_${idtgtod}${HR}_${idtg}${HR}.com
```

```
#
# --- Submit job to transfer observations files for NCODA
#
/u/home/wallcraf/bin/q_navo ${E}ncoda_PREP_${idtgtod}${HR}_${idtg}${HR}.csh
exit
```

APPENDIX B

Script 727lsf_wind_prep_${idtgtod}18_${idtg}18.com

This is the atmospheric forcing generation script.

```
#!/bin/csh -f
#PBS -N 999pbs
#PBS -j oe
#PBS -o 999pbsXX.log
#PBS -l mppwidth=1
#PBS -l mppnppn=1
#PBS -l walltime=3:00:00
#PBS -W umask=027
#PBS -A NRLSS018
#PBS -q internal3d
#
#BSUB -J 727ncodawind
#BSUB -n 1
#BSUB -R "span[ptile=1]"
#BSUB -W 4:00
#BSUB -q share
#BSUB -P NAVOSHYC
#
# --- Job to create the forcing needed for a HYCOM-NCODA run
#
set echo
set timestamp
C
C --- Preamble.
C
setenv OS `uname`
switch ($OS)
case 'Linux':
    which yod
    if (! $status) then
      setenv OS XT3
      setenv TMPDIR /tmp
    else
      setenv TMPDIR /tmp
    endif
    which aprun
    if (! $status) then
#     setenv OS XT4
      setenv OS XT5
      setenv TMPDIR /scr
    endif
    breaksw
case 'AIX':
    hostname
    setenv TMPDIR /scr/${user}
    breaksw
default:
    echo 'Unknown Operating System: ' $OS
```

```
    echo 'configured for AIX only'
    exit (1)
endsw
C
if ($?JOBNAME) then
    setenv PBS_JOBNAME ${JOBNAME}
    setenv PBS_JOBID   $$
endif
echo PBS_JOBNAME $PBS_JOBNAME PBS_JOBID $PBS_JOBID
C
C --- E is expt, P is permenant directory, S is /tmp directory.
C
#
setenv E 727
setenv EXPT expt_72.7
setenv REG GLBa0.08
#
# --- These variables are set by the awk script
#
setenv nmdays 10
setenv idtg 20070209
setenv idtg2 `/u/home/${user}/bin/addndays yyyymmdd ${idtg} +1`
setenv idtgtod 20070214
setenv HR 00
setenv HR2 00
#
echo ${idtg} ${idtgtod}
#
set LOGS=/scr/${user}/hycom/${REG}/${EXPT}/logs
mkdir -p ${LOGS}
set RUN=/u/home/${user}/hycom/${REG}/${EXPT}
cd ${RUN}
#
setenv P $cwd
switch ($OS)
case 'AIX':
case 'XT5':
#
# --- substitute /scr for /u/home
#
    setenv S `echo $cwd | awk '{print "/scr" substr($0,8,length)}'`
    breaksw
case 'XT3':
case 'XT4':
case 'OSF1':
#                           substitute /work for /???????
    setenv S `echo $cwd | awk '{print "/work" substr($0,3,length)}'`
    breaksw
case 'IRIX64':
#                           substitute /scr for /u/home
    setenv S `echo $cwd | awk '{print "/scr" substr($0,8,length)}'`
    breaksw
case 'unicos':
#                           substitute /tmp for /u/b
    setenv S `echo $cwd | awk '{print "/tmp" substr($0,5,length)}'`
    breaksw
endsw
C
```

```
ls -laFq
set idtgtod1=`echo ${idtgtod} | cut -c1-8`
C
C --- check the RUNNING flag.
C
if ( -e RUNNING && ! -e RUNNING_$PBS_JOBID) then
C
C --- MODEL IS ALREADY RUNNING - EXIT.
C
  exit
endif
touch RUNNING
touch RUNNING_$PBS_JOBID
C
C --- Generate the next model script.
C
  setenv Y01 103
  setenv AB a
#
  setenv SCRIPT ${E}y${idtg}.com
  /bin/rm -f ${SCRIPT}
  awk -f ${RUN}/${E}.awk nmdays=${nmdays} y01=${Y01} hr=${HR} tod=${idtgtod} \
   ab=${AB} td=${idtg} ${RUN}/${E}_wind.com > ${SCRIPT}
#
# --- Run the Script.
#
set script = $SCRIPT
set reqname = ${PBS_JOBNAME}
ln -fs ${reqname}.log $script:r.log
#
mkdir -p $S
cp  ${SCRIPT} $S/${SCRIPT}
cd  $S
#
# -------------------------------------------------------------------------
csh ${SCRIPT}
# -------------------------------------------------------------------------
#
ls -laFq
#
cd  $P
#
# --- Final Clean Up.
#
/bin/rm -f RUNNING
/bin/rm -f RUNNING_$PBS_JOBID
#
# --- Make links for the forcing files created today
#
while(${idtg2} <= ${idtgtod1})
#
cd /scr/${user}/hycom/${REG}/${EXPT}/data/wind
/bin/rm *${idtg2}.*
ln -s tauewd_${idtg}.a tauewd_${idtg2}.a
ln -s tauewd_${idtg}.b tauewd_${idtg2}.b
ln -s taunwd_${idtg}.a taunwd_${idtg2}.a
ln -s taunwd_${idtg}.b taunwd_${idtg2}.b
#
```

```
cd /scr/${user}/hycom/${REG}/${EXPT}/data/flux
/bin/rm *${idtg2}.*
ln -s airtmp_${idtg}.a airtmp_${idtg2}.a
ln -s airtmp_${idtg}.b airtmp_${idtg2}.b
ln -s radflx_${idtg}.a radflx_${idtg2}.a
ln -s radflx_${idtg}.b radflx_${idtg2}.b
ln -s shwflx_${idtg}.a shwflx_${idtg2}.a
ln -s shwflx_${idtg}.b shwflx_${idtg2}.b
ln -s vapmix_${idtg}.a vapmix_${idtg2}.a
ln -s vapmix_${idtg}.b vapmix_${idtg2}.b

cd /scr/${user}/hycom/${REG}/${EXPT}/data/ssta
/bin/rm *${idtg2}.*
ln -s surtmp_${idtg}.a surtmp_${idtg2}.a
ln -s surtmp_${idtg}.b surtmp_${idtg2}.b

cd /scr/${user}/hycom/${REG}/${EXPT}/data/pcip
/bin/rm *${idtg2}.*
ln -s precip_${idtg}.a precip_${idtg2}.a
ln -s precip_${idtg}.b precip_${idtg2}.b

cd /scr/${user}/hycom/${REG}/${EXPT}/data/wspd
/bin/rm *${idtg2}.*
ln -s wndspd_${idtg}.a wndspd_${idtg2}.a
ln -s wndspd_${idtg}.b wndspd_${idtg2}.b
#
set idtg2=`/u/home/${user}/bin/addndays yyyymmdd ${idtg2} +1`
end
cd ${LOGS}
#
# --- Submit the job extracting an archive file from a restart file.
# --- The job will wait for the time slot dedicated to the HYCOM run
#
#/u/home/wallcraf/bin/q_navo ${E}rest2archv_${idtgtod}${HR}_${idtg}${HR}.csh
/u/home/wallcraf/bin/q_navo ${E}ncoda_PREP_${idtgtod}${HR}_${idtg}${HR}.csh
#
# --- Exit.
#
exit
```

APPENDIX C

Script 727extr_${subreg}_${idtgtod}18_${idtg}18.com

This script extracts NCODA z-levels from the HYCOM archive file for each NCODA subregion.

```
#!/bin/csh -f
#PBS -N 999pbs
#PBS -j oe
#PBS -o 999pbsXX.log
#PBS -l mppwidth=1
#PBS -l mppnppn=1
#PBS -l walltime=1:00:00
#PBS -W umask=027
#PBS -A NRLSS018
#PBS -q internal3d
#
#BSUB -J pmast
#BSUB -n 1
#BSUB -W 0:30
#BSUB -R "span[ptile=1]"
#BSUB -q challenge
#BSUB -P NRLSSC3J
#####BSUB -q internal3d
####BSUB -b 10:24:20:00
#
date
set echo
set pget=~wallcraf/bin/pget
#
set OS=`uname`
switch ($OS)
case 'Linux':
    which aprun
    if (! $status) then
      set APRUN='aprun -n 1 '
      set SRC=~wallcraf/hycom/ALLcnl
    else
      set APRUN=''
      set SRC=~wallcraf/hycom/ALL
    endif
 breaksw
case 'AIX'
 set APRUN=''
 set SRC=~wallcraf/hycom/ALL
 breaksw
default:
 set APRUN=''
 set SRC=~wallcraf/hycom/ALL
endsw
#
# --- extract 3-d fields from a single HYCOM archive file.
#
setenv CALNMB 1
setenv HR 18
setenv HR2 18
setenv idtg 20080501
```

```
setenv idtgtod 20080501
setenv idtgtodp1 20080501
setenv idtgtodm1 20080501
set idtgout=`/u/home/${user}/bin/addndays yyyymmdd ${idtg} -1`
#
echo 'DATES TO EXTRACT ' ${idtg}
#
set EXPTN=expt_72.7
set REGN=MERpa10.08
#
setenv E 727
set EXPT=expt_72.7
set REG=GLBa0.08
set CASE=nowcast
setenv T 09
#
set maxdepth=002500
set surface=000000
set forecast=00240000
#
set TOPO=/u/home/${user}/hycom/${REG}/topo
set SCR=/scr/${user}/hycom/${REG}/${EXPT}
set LOGS=/scr/${user}/hycom/${REG}/${EXPT}/logs
set SCRN=/scr/${user}/hycom/${REGN}/${EXPTN}
mkdir -p /scr/${user}/hycom/${REG}/topo
mkdir -p ${SCR} ${SCRN}
set INP=${SCR}/${CASE}
set OUT=${SCRN}/raw/${CASE}
mkdir -p ${OUT}
echo ${SCR} ${SCRN}
#
cd ${OUT}
/bin/rm ${OUT}/*[ab]
#
goto TOPO
touch ${TOPO}/regional.depth.a ${TOPO}/regional.depth.b
if (-z ${TOPO}/regional.depth.a) then
   ${pget} ${D}/../../topo/depth_${REG}_${T}.a ${TOPO}/regional.depth.a &
endif
if (-z ${TOPO}/regional.depth.b) then
   ${pget} ${D}/../../topo/depth_${REG}_${T}.b ${TOPO}/regional.depth.b &
endif
touch ${TOPO}/regional.grid.a ${TOPO}/regional.grid.b
if (-z ${TOPO}/regional.grid.a) then
   ${pget} ${D}/../../topo/regional.grid.a ${TOPO}/regional.grid.a &
endif
if (-z ${TOPO}/regional.grid.b) then
   ${pget} ${D}/../../topo/regional.grid.b ${TOPO}/regional.grid.b &
endif
touch ${TOPO}/regional_mask.a
if (-z ${TOPO}/regional_mask.a) then
   ${pget} ${D}/../../topo/landsea_${REG}.a ${TOPO}/regional_mask.a &
endif
wait
TOPO:
#
#ln -s ${TOPO}/depth_${REG}_${T}.a regional.depth.a
#ln -s ${TOPO}/depth_${REG}_${T}.b regional.depth.b
#ln -s ${TOPO}/regional.grid.a .
#ln -s ${TOPO}/regional.grid.b .
#ln -s ${TOPO}/regional_mask.a .
#
ln -s ${INP}/../data/regional.depth.a .
```

```
ln -s ${INP}/../data/regional.depth.b .
ln -s ${INP}/../data/regional.grid.a .
ln -s ${INP}/../data/regional.grid.b .
#/bin/cp ${TOPO}/regional_mask.a .
#
#/bin/cp ${TOPO}/depth_${REG}_${T}.a regional.depth.a
#/bin/cp ${TOPO}/depth_${REG}_${T}.b regional.depth.b
#/bin/cp ${TOPO}/regional.grid.a .
#/bin/cp ${TOPO}/regional.grid.b .
#/bin/cp ${TOPO}/regional_mask.a .
#
    setenv FOR022A ${OUT}/hycom2dath_000000_${idtgout}${HR}_${forecast}.a
    setenv FOR022  ${OUT}/hycom2dath_000000_${idtgout}${HR}_${forecast}.b
    setenv FOR033A  ${OUT}/hycom3dt_${maxdepth}_${idtgout}${HR}_${forecast}.a
    setenv FOR033  ${OUT}/hycom3dt_${maxdepth}_${idtgout}${HR}_${forecast}.b
    setenv FOR034A  ${OUT}/hycom3ds_${maxdepth}_${idtgout}${HR}_${forecast}.a
    setenv FOR034  ${OUT}/hycom3ds_${maxdepth}_${idtgout}${HR}_${forecast}.b
#    setenv FOR035A  ${OUT}/hycom3dden_${maxdepth}_${idtgout}${HR}_${forecast}.a
#    setenv FOR035  ${OUT}/hycom3dden_${maxdepth}_${idtgout}${HR}_${forecast}.b
    setenv FOR037A  ${OUT}/hycom3du_${maxdepth}_${idtgout}${HR}_${forecast}.a
    setenv FOR037  ${OUT}/hycom3du_${maxdepth}_${idtgout}${HR}_${forecast}.b
    setenv FOR038A  ${OUT}/hycom3dv_${maxdepth}_${idtgout}${HR}_${forecast}.a
    setenv FOR038  ${OUT}/hycom3dv_${maxdepth}_${idtgout}${HR}_${forecast}.b
#    setenv FOR040A  ${OUT}/hycom3dp_${maxdepth}_${idtgout}${HR}_${forecast}.a
#    setenv FOR040  ${OUT}/hycom3dp_${maxdepth}_${idtgout}${HR}_${forecast}.b
    /bin/rm $FOR033A $FOR034A $FOR037A $FOR038A
    /bin/rm $FOR033 $FOR034 $FOR037 $FOR038
    /bin/rm $FOR022
    /bin/rm $FOR022A
#
${APRUN} ${SRC}/archive/src/archv2data3z <<E-o-D
${INP}/archv.${idtgtodm1}${HR}_${idtg}${HR2}.a
HYCOM
000     'iexpt ' = experiment number x10 (000=from archive file)
  3     'yrflag' = days in year flag (0=360J16,1=366J16,2=366J01,3-actual)
4500     'idm   ' = longitudinal array size
3298     'jdm   ' = latitudinal  array size
 32      'kdm   ' = number of layers
 34.0   'thbase' = reference density (sigma units)
   0   'smooth' = smooth the layered fields (0=F,1=T)
   1    'baclin' = extract baroclinic velocity (0=total,1=baroclinic)
   1     'xyward' = output original unrotated velocities (0=no:DEFAULT,1=yes)
199    'iorign' = i-origin of plotted subregion
360    'jorign' = j-origin of plotted subregion
2525   'idmp  ' = i-extent of plotted subregion (<=idm; 0 implies idm)
1841   'jdmp  ' = j-extent of plotted subregion (<=jdm; 0 implies jdm)
   2    'itype ' = interpolation type (0=sample,1=linear,2=parabolic)
   42  'kzi    ' = number of depths to sample
   0.0   'zi     ' = sample cell interfaces  1
   3.5   'zi     ' = sample cell interfaces  2
   6.5   'zi     ' = sample cell interfaces  3
   9.5   'zi     ' = sample cell interfaces  4
  14.5   'zi     ' = sample cell interfaces  5
  19.5   'zi     ' = sample cell interfaces  6
  30.5   'zi     ' = sample cell interfaces  7
  41.5   'zi     ' = sample cell interfaces  8
  58.5   'zi     ' = sample cell interfaces  9
  87.5   'zi     ' = sample cell interfaces 10
 112.5   'zi     ' = sample cell interfaces 11
 137.5   'zi     ' = sample cell interfaces 12
 162.5   'zi     ' = sample cell interfaces 13
 187.5   'zi     ' = sample cell interfaces 14
 212.5   'zi     ' = sample cell interfaces 15
```

```
 237.5   'zi     ' = sample cell interfaces 16
 262.5   'zi     ' = sample cell interfaces 17
 287.5   'zi     ' = sample cell interfaces 18
 312.5   'zi     ' = sample cell interfaces 19
 375.0   'zi     ' = sample cell interfaces 20
 425.0   'zi     ' = sample cell interfaces 21
 475.0   'zi     ' = sample cell interfaces 22
 525.0   'zi     ' = sample cell interfaces 23
 650.0   'zi     ' = sample cell interfaces 24
 750.0   'zi     ' = sample cell interfaces 25
 850.0   'zi     ' = sample cell interfaces 26
 950.0   'zi     ' = sample cell interfaces 27
1050.0   'zi     ' = sample cell interfaces 28
1150.0   'zi     ' = sample cell interfaces 29
1250.0   'zi     ' = sample cell interfaces 30
1350.0   'zi     ' = sample cell interfaces 31
1450.0   'zi     ' = sample cell interfaces 32
1550.0   'zi     ' = sample cell interfaces 33
1650.0   'zi     ' = sample cell interfaces 34
1750.0   'zi     ' = sample cell interfaces 35
1850.0   'zi     ' = sample cell interfaces 36
1950.0   'zi     ' = sample cell interfaces 37
2050.0   'zi     ' = sample cell interfaces 38
2150.0   'zi     ' = sample cell interfaces 39
2250.0   'zi     ' = sample cell interfaces 40
2350.0   'zi     ' = sample cell interfaces 41
2450.0   'zi     ' = sample cell interfaces 42
2550.0   'zi     ' = sample cell interfaces 43
  0    'botio ' = bathymetry  I/O unit (0 no I/O)
  22    'athio'  = average density  I/O unit (0 no I/O)
  0    'mltio ' = mix.l.thk.  I/O unit (0 no I/O)
  0.0   'tempml' = temperature jump across mixed-layer (degC,  0 no I/O)
  0.0   'densml' =     density jump across mixed-layer (kg/m3, 0 no I/O)
  0    'infio ' = intf. depth I/O unit (0 no I/O, <0 label with layer #)
  0     'wvlio ' = w-velocity       I/O unit (0 no I/O)
  37   'uvlio ' = u-velocity  I/O unit (0 no I/O)
  38   'vvlio ' = v-velocity  I/O unit (0 no I/O)
   0    'splio ' = speed       I/O unit (0 no I/O)
  33    'temio ' = temperature I/O unit (0 no I/O)
  34    'salio ' = salinity    I/O unit (0 no I/O)
  0     'tthio ' = density     I/O unit (0 no I/O)
E-o-D
#
#/usr/lpp/LoadL/full/bin/llq -w $LOADL_STEP_ID
#
# --- convert HYCOM .a files to RAW files (no padding, spval=1.e10).
# --- comment this out if you don't need RAW files.
#
    foreach t ( 3dt 3ds 3du 3dv )
      if (-e ${OUT}/hycom${t}_${maxdepth}_${idtgout}${HR}_${forecast}.a) then
        /bin/rm -f ${OUT}/hycom${t}_${maxdepth}_${idtgout}${HR}_${forecast}.A
        ${APRUN} ${SRC}/bin/hycom2raw
${OUT}/hycom${t}_${maxdepth}_${idtgout}${HR}_${forecast}.a 2525 1841 -999.0
${OUT}/hycom${t}_${maxdepth}_${idtgout}${HR}_${forecast}.A
      endif
    end
#
cd ${OUT}
#
# --- extract 2-d fields from a single HYCOM archive file.
#
set idtgin=${idtg}
#    foreach HR2 ( 21 00 03 06 09 12 15 18 )
```

```
    foreach HR2 ( 18 )
if( ${HR2} == 21 ) then
 set forecast=00030000
 set idtgin=`/u/home/${user}/bin/addndays yyyymmdd ${idtg} -1`
 if( ${CALNMB} == 1 ) then
  set idtgtodin=`/u/home/${user}/bin/addndays yyyymmdd ${idtgtod} -1`
 else
  set idtgtodin=${idtgtod}
 endif
endif
if( ${HR2} == 00 ) then
 set forecast=00060000
 set idtgin=${idtg}
 if( ${CALNMB} == 1 ) then
  set idtgtodin=`/u/home/${user}/bin/addndays yyyymmdd ${idtgtod} -1`
 else
  set idtgtodin=${idtgtod}
 endif
endif
if( ${HR2} == 03 ) then
 set forecast=00090000
 set idtgin=${idtg}
 if( ${CALNMB} == 1 ) then
  set idtgtodin=`/u/home/${user}/bin/addndays yyyymmdd ${idtgtod} -1`
 else
  set idtgtodin=${idtgtod}
 endif
endif
if( ${HR2} == 06 ) then
 set forecast=00120000
 set idtgin=${idtg}
 if( ${CALNMB} == 1 ) then
  set idtgtodin=`/u/home/${user}/bin/addndays yyyymmdd ${idtgtod} -1`
 else
  set idtgtodin=${idtgtod}
 endif
endif
if( ${HR2} == 09 ) then
 set forecast=00150000
 set idtgin=${idtg}
 if( ${CALNMB} == 1 ) then
  set idtgtodin=`/u/home/${user}/bin/addndays yyyymmdd ${idtgtod} -1`
 else
  set idtgtodin=${idtgtod}
 endif
endif
if( ${HR2} == 12 ) then
 set forecast=00180000
 set idtgin=${idtg}
 if( ${CALNMB} == 1 ) then
  set idtgtodin=`/u/home/${user}/bin/addndays yyyymmdd ${idtgtod} -1`
 else
  set idtgtodin=${idtgtod}
 endif
endif
if( ${HR2} == 15 ) then
 set forecast=00210000
 set idtgin=${idtg}
 if( ${CALNMB} == 1 ) then
  set idtgtodin=`/u/home/${user}/bin/addndays yyyymmdd ${idtgtod} -1`
 else
  set idtgtodin=${idtgtod}
 endif
```

```
endif
if( ${HR2} == 18 ) then
 set forecast=00240000
 set idtgin=${idtg}
 set idtgtodin=${idtgtodm1}
endif
#
#    setenv FOR022A  ${OUT}/hycom2dSfsd_${surface}_${idtgout}${HR}_${forecast}.a
#    setenv FOR022  ${OUT}/hycom2dSfsd_${surface}_${idtgout}${HR}_${forecast}.b
    setenv FOR023A  ${OUT}/hycom2dfsd_${surface}_${idtgout}${HR}_${forecast}.a
    setenv FOR023  ${OUT}/hycom2dfsd_${surface}_${idtgout}${HR}_${forecast}.b
#    setenv FOR024A  ${OUT}/hycom2dNfsd_${surface}_${idtgout}${HR}_${forecast}.a
#    setenv FOR024  ${OUT}/hycom2dNfsd_${surface}_${idtgout}${HR}_${forecast}.b
    setenv FOR025A  ${OUT}/hycom2dt_${surface}_${idtgout}${HR}_${forecast}.a
    setenv FOR025  ${OUT}/hycom2dt_${surface}_${idtgout}${HR}_${forecast}.b
    setenv FOR026A  ${OUT}/hycom2dice_${surface}_${idtgout}${HR}_${forecast}.a
    setenv FOR026  ${OUT}/hycom2dice_${surface}_${idtgout}${HR}_${forecast}.b
    setenv FOR027A  ${OUT}/hycom2dmix_${surface}_${idtgout}${HR}_${forecast}.a
    setenv FOR027  ${OUT}/hycom2dmix_${surface}_${idtgout}${HR}_${forecast}.b
    /bin/rm $FOR023A $FOR025A $FOR026A $FOR027A
    /bin/rm $FOR023 $FOR025 $FOR026 $FOR027
#
${APRUN} ${SRC}/archive/src/archv2data2d <<E-o-D
${INP}/archv.${idtgtodin}${HR}_${idtgin}${HR2}.a
HYCOM
000     'iexpt ' = experiment number x10 (000=from archive file)
  3     'yrflag' = days in year flag (0=360J16,1=366J16,2=366J01,3=actual)
4500     'idm   ' = longitudinal array size
3298     'jdm   ' = latitudinal  array size
 1      'kdm   ' = number of layers
34.0   'thbase' = reference density (sigma units)
  0    'smooth' = smooth fields before plotting (0=F,1=T)
  0    'mthin ' = mask thin layers from plots   (0=F,1=T)
  1     'xyward' = output original unrotated velocities (0=no:DEFAULT,1=yes)
199     'iorign' = i-origin of plotted subregion
360     'jorign' = j-origin of plotted subregion
2525    'idmp  ' = i-extent of plotted subregion (<=idm; 0 implies idm)
1841    'jdmp  ' = j-extent of plotted subregion (<=jdm; 0 implies jdm)
 0     'botio ' = bathymetry       I/O unit (0 no I/O)
  0    'flxio ' = surf. heat flux  I/O unit (0 no I/O)
  0    'empio ' = surf. evap-pcip  I/O unit (0 no I/O)
  0    'ttrio ' = surf. temp trend I/O unit (0 no I/O)
  0    'strio ' = surf. saln trend I/O unit (0 no I/O)
 26    'icvio ' = ice coverage     I/O unit (0 no I/O)
  0    'ithio ' = ice thickness    I/O unit (0 no I/O)
  0    'ictio ' = ice temperature  I/O unit (0 no I/O)
 23    'sshio ' = sea surf. height I/O unit (0 no I/O)
  0    'bsfio ' = baro. strmfn.    I/O unit (0 no I/O)
 0     'uvmio ' = mix. lay. u-vel. I/O unit (0 no I/O)
 0     'vvmio ' = mix. lay. v-vel. I/O unit (0 no I/O)
  0    'spmio ' = mix. lay. speed  I/O unit (0 no I/O)
 0     'bltio ' = bnd. lay. thick. I/O unit (0 no I/O)
 27    'mltio ' = mix. lay. thick. I/O unit (0 no I/O)
 0     'sstio ' = mix. lay. temp.  I/O unit (0 no I/O)
 0     'sssio ' = mix. lay. saln.  I/O unit (0 no I/O)
 0     'ssdio ' = mix. lay. dens.  I/O unit (0 no I/O)
-1      'kf     ' = layer to plot (=0 end layer plots; <0 label with layer #)
 1     'kl     ' = last  output layer
 0     'uvlio ' = layer k   u-vel. I/O unit (0 no I/O)
 0     'vvlio ' = layer k   v-vel. I/O unit (0 no I/O)
  0     'splio ' = layer k   speed. I/O unit (0 no I/O)
 0     'infio ' = layer k   i.dep. I/O unit (0 no I/O)
 0     'thkio ' = layer k   thick. I/O unit (0 no I/O)
```

```
 25     'temio ' = layer k   temp   I/O unit (0 no I/O)
 0     'salio ' = layer k   saln.  I/O unit (0 no I/O)
 0     'tthio ' = layer k   dens,  I/O unit (0 no I/O)
 0     'sfnio ' = layer k  strmfn. I/O unit (0 no I/O)
 0     'kf    ' = first output layer (=0 end output; <0 label with layer #)
E-o-D
#
# convert ice concentration to percent
#
/bin/rm hycom2dice_${surface}_${idtgout}${HR}_${forecast}.a2
 ${APRUN} ${SRC}/bin/hycom_expr
hycom2dice_${surface}_${idtgout}${HR}_${forecast}.a ONE 2525 1841 100.0 0.0
hycom2dice_${surface}_${idtgout}${HR}_${forecast}.a2
/bin/mv hycom2dice_${surface}_${idtgout}${HR}_${forecast}.a2
hycom2dice_${surface}_${idtgout}${HR}_${forecast}.a
#
    foreach t ( fsd t ice mix)
      if (-e ${OUT}/hycom2d${t}_${surface}_${idtgout}${HR}_${forecast}.a) then
        /bin/rm -f ${OUT}/hycom2d${t}_${surface}_${idtgout}${HR}_${forecast}.A
         ${APRUN} ${SRC}/bin/hycom2raw
${OUT}/hycom2d${t}_${surface}_${idtgout}${HR}_${forecast}.a 2525 1841 -999.0
${OUT}/hycom2d${t}_${surface}_${idtgout}${HR}_${forecast}.A
      endif
    end
#
#/usr/lpp/LoadL/full/bin/llq -w $LOADL_STEP_ID
#
# calculate steric ssh
#
set MEAN=094_archMNA.0011_0015_ATH_${REGN}.a
/bin/cp /u/home/${user}/hycom/${REG}/meanstd/094_archMNA.0011_0015_ATH_${REGN}.a
.
#
foreach a ( 000000_${idtgout}${HR}_${forecast} )
  /bin/rm -f hycom2d[NS]fsd*.?
   ${APRUN} ${SRC}/bin/hycom_stericssh hycom2dath_${a}.a ${MEAN}  2525 1841
hycom2dSfsd_${a}.a >! hycom2dSfsd_${a}.b
   ${APRUN} ${SRC}/bin/hycom_expr      hycom2dfsd_${a}.a hycom2dSfsd_${a}.a 2525
1841 1.0 -1.0 hycom2dNfsd_${a}.a >! hycom2dNfsd_${a}.b
end
#
    foreach t ( Nfsd Sfsd)
      if (-e ${OUT}/hycom2d${t}_${surface}_${idtgout}${HR}_${forecast}.a) then
        /bin/rm -f ${OUT}/hycom2d${t}_${surface}_${idtgout}${HR}_${forecast}.A
         ${APRUN} ${SRC}/bin/hycom2raw
${OUT}/hycom2d${t}_${surface}_${idtgout}${HR}_${forecast}.a 2525 1841 -999.0
${OUT}/hycom2d${t}_${surface}_${idtgout}${HR}_${forecast}.A
      endif
    end
#
# forecash HR2
end
#
date
#
# submit ncoda analysis
#
cd ${LOGS}
~wallcraf/bin/q_navo ${E}ncoda_MERpa1_${idtgtodp1}${HR}_${idtg}${HR}.csh
```

APPENDIX D

Script 727nocda_${subreg}_${idtgtod}18_${idtg}18.com

This is a script to run the NCODA analysis for each subregion.

```
#!/bin/ksh
#PBS -N 999pbs
#PBS -j oe
#PBS -o 999pbsXX.log
#PBS -l mppwidth=416
#PBS -l mppnppn=8
#PBS -l walltime=4:00:00
#PBS -W umask=027
#PBS -A NRLSS018
#PBS -q internal3d
#
#BSUB -J 727ncoda_MERpa1.job
#BSUB -n 192
#BSUB -a poe
#BSUB -W 3:30
#BSUB -R "rusage[ntbl_windows=32]span[ptile=16]"
#BSUB -q challenge
#BSUB -P NRLSSC3J
#####BSUB -q internal3d
#####BSUB -P NAVOSHYC
#BSUB -e err_MERpa1.%J
#
# --- set days to run
#
inumd=1
#
# --- analysis date
#
idtg=2008050118
#
# --- forecast from this date
#
ifrcst=2008043018
#
# --- day when this run is done (just in output and script being submitted)
#
idtgtod=2008050118
#
# Change note:
# Removed the TMPDIR exclusion.

### @ environment      = COPY_ALL; !TMPDIR;

# ------------------------------------------------------------
#       Script:   ncoda_MERpa1.job
#       Purpose:  to setup and cycle the CODA analysis only
#                 in the HYCOM 1/12 deg Gulf of Mexico grid
#       Author:   Steve Lowder, NRL-CSC
#                 hacked by Jim Cummings
#       Created:  1 Mar 2004
#       Platform: IBM SP5 at NAVO
#       Usage:    see usage() function, below
# ------------------------------------------------------------
```

```
# ------------------------------------------------------
# DATA NOTE:
# This script copies data from a mass storage server to
# a local archive area (scratch). When the analysis
# runs it operates in another run directory (scratch). At the
# end of the run, this script does not move the results to
# a mass storage server. The move to the permanent storage
# is done with the plotting script submitted at the end of the script.
# ------------------------------------------------------

# ------------------------------------------------------
# FUNCTIONS
# Most of the work in this script is organized into
# functions which are defined at the beginning of the
# script. Search for "Begin script" to skip them.
# ------------------------------------------------------

function usage {
    print ncoda_MERpa1.job
    print "usage: ncoda_MERpa1.job start_dtg num_days frcst_dtg"
    print "       start_dtg is the first day in yyyymmddhh format"
    print "       num_days  is the total number of days to cycle"
    print "       frcst_dtg is the forecast day in yyyymmddhh format"
}

function fatal {
    echo " "
    echo " "
    echo "====================================================="
    echo " "
    echo " "
    echo "Script stopped on error condition at " `date +%T`
    echo " "
    echo " "
    echo "====================================================="
    exit 1
}

function setup_paths {
#
# There are three logical areas to this script.
# 1. The mass storage areas where data is permanently stored.
# 2. The archive area on the computer where mass storage data is
#    copied prior to a run.
# 3. The run area on the computer where the application does its
#    work.
# Setup paths to the mass storage server
#

# SAL change:
    T=09
    E=727
    EXPT=expt_72.7
    REG=MERpa10.08
    REG1=MERpa1
    REG0=GLBa0.08
    MSASv=newton
    MSAS_LOCA_MDL=/scr/${USER}/hycom/${REG}/${EXPT}/raw/nowcast
    MSAS_WORK_TOP=/u/home/${USER}/hycom/${REG0}/subtopo/${REG1}
    MSAS_WORK_MDL=/u/home/${USER}/hycom/${REG0}/meanstd
    MSAS_WORK_OBS=${MSASv}:/u/home/ooc/data/ncoda/ocnqc
#
# Setup paths on the computer in the user's scratch and home directory
```

```
#
# BIN_DIR       - the location of all executables
# SCRATCH_DIR   - the location of user's scratch area
# MODEL_DIR     - the location of model specific static data
# DB_DIR        - the location of static database files like clim,land-sea
# OBS_DIR       - the location of ocean data observations like ship, ssmi
#
  BIN_DIR=/u/home/jac/ncoda_mpi/bin
  DDTG=/u/home/jac/ncoda_dtg/bin/dtg
#  BIN_DIR=/u/home/ooc/models/ncoda/ncoda_mpi/bin
#  DDTG=/u/home/ooc/models/ncoda/ncoda_dtg/bin/dtg
  SCRATCH_DIR=/scr/$USER
  RUN_DIR=$SCRATCH_DIR/${REG}/${EXPT}
  SAVE_DIR=$SCRATCH_DIR/${REG0}/ncoda/${EXPT}
  RESTART_DIR=$RUN_DIR/restart
  DB_DIR=$SCRATCH_DIR/database
  OBS_DIR=$DB_DIR/ocnqc
}

function setup_dirs {
    cd $SCRATCH_DIR

    export TMPDIR=$SCRATCH_DIR/tmp
    mkdir -p $TMPDIR

#
#   Make the static database dir, obs, dir, and model dir if
#   they do not exist.
#

    mkdir -p $DB_DIR
    mkdir -p ${DB_DIR}/clim
    mkdir -p ${DB_DIR}/gdem


    mkdir -p $OBS_DIR
    mkdir -p ${OBS_DIR}/beta
    cd ${OBS_DIR}/beta
    mkdir -p altim
    mkdir -p goes
    mkdir -p lac
    mkdir -p profile
    mkdir -p ship
    mkdir -p ssmi
    mkdir -p ${OBS_DIR}/gamma
    cd ${OBS_DIR}/gamma
    mkdir -p altim
    mkdir -p goes
    mkdir -p lac
    mkdir -p profile
    mkdir -p ship
    mkdir -p ssmi
    mkdir -p ${OBS_DIR}/kappa
    cd ${OBS_DIR}/kappa
    mkdir -p altim
    mkdir -p goes
    mkdir -p lac
    mkdir -p profile
    mkdir -p ship
    mkdir -p ssmi
    mkdir -p ${OBS_DIR}/godae
    cd ${OBS_DIR}/godae
    mkdir -p altim
```

```
    mkdir -p mcsst
    mkdir -p profile
    mkdir -p ship
    mkdir -p ssmi
#
# Now remake the run directory to be sure it is clean.
#
#  rm -rf   $RUN_DIR/analysis
#  rm -rf   $RUN_DIR/restart
#  mkdir -p $RUN_DIR
  mkdir -p $RUN_DIR/analysis
  mkdir -p $RUN_DIR/restart
  mkdir -p $RUN_DIR/output
  mkdir -p $SAVE_DIR
}

function setup_data {
#
# Define local variables
#
    typeset last_dtg
    typeset -i n_back f_frwd
#
# Prepare the data for the entire run. You will need data from
# three logical areas:
# 1. Static data   2. Observation data   3. Model area data
#
# Build or replenish the Model area data
#
    echo " "
    echo "=================================================="
    echo " "

#
# Set the parameters for the number of days forward
# and backward to look for obs.
#
    let n_back=-10
    let n_frwd=2+${inumd}
#
    print -n "      Checking depth file (date time group indpendent)"
    cd ${RESTART_DIR}
#    if [ ! -r depths_sfc_000000_000000_1o2525x1841_${prev_dtg}_00000000_datafld
]; then
       /bin/cp $MSAS_WORK_TOP/depth_${REG}_${T}.A
depths_sfc_000000_000000_1o2525x1841_${prev_dtg}_00000000_datafld
#    fi
    echo "..... ready"

    print -n "      Checking initial model error file (should be name with
initial date)"
    cd ${RESTART_DIR}
#    if [ ! -r seahgt_sfc_000000_000000_1o2525x1841_1999080900_00000000_modlerr
]; then
#      rcp
$MSAS_WORK_MDL/seahgt_sfc_000000_000000_1o2525x1841_1999080900_00000000_modlerr .
#      rcp
$MSAS_WORK_MDL/seatmp_sfc_000000_000000_1o2525x1841_1999080900_00000000_modlerr .
#      rcp
$MSAS_WORK_MDL/seatmp_pre_000000_002500_1o2525x1841_1999080900_00000000_modlerr .
#      rcp
$MSAS_WORK_MDL/salint_pre_000000_002500_1o2525x1841_1999080900_00000000_modlerr .
```

```
#      rcp
$MSAS_WORK_MDL/uucurr_pre_000000_002500_1o2525x1841_1999080900_00000000_modlerr .
#      rcp
$MSAS_WORK_MDL/vvcurr_pre_000000_002500_1o2525x1841_1999080900_00000000_modlerr .
#    fi
#    echo "..... ready"

   FCST=${ifrcst}
   echo 'FORECAST DATE '${FCST}

    print -n "      Checking first guess fields valid tau 024"
    cd ${RESTART_DIR}
#    if [ ! -r seatmp_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld ];
then
#      rcp $MSAS_WORK_MDL/hycom2dt_000000_${FCST}_00240000.A
seatmp_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld
       /bin/cp $MSAS_LOCA_MDL/hycom2dt_000000_${FCST}_00240000.A
seatmp_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld
#    fi
#    if [ ! -r seahgt_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld ];
then
#      rcp $MSAS_WORK_MDL/hycom2dSfsd_000000_${FCST}_00240000.A
seahgt_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld
       /bin/cp $MSAS_LOCA_MDL/hycom2dSfsd_000000_${FCST}_00240000.A
seahgt_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld
#for tau in 00030000 00060000 00090000 00120000 00150000 00180000 00210000
00240000
#do
#       /bin/cp $MSAS_LOCA_MDL/hycom2dSfsd_000000_${FCST}_${tau}.A
seahgt_sfc_000000_000000_1o2525x1841_${FCST}_${tau}_fcstfld
#done
#    fi
#    if [ ! -r seaice_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld ];
then
#        rcp $MSAS_WORK_MDL/hycom2dice_000000_${FCST}_00240000.A
seaice_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld
        /bin/cp $MSAS_LOCA_MDL/hycom2dice_000000_${FCST}_00240000.A
seaice_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld
#    fi
#    if [ ! -r mixlyr_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld ];
then
#        rcp $MSAS_WORK_MDL/hycom2dmix_000000_${FCST}_00240000.A
mixlyr_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld
        /bin/cp $MSAS_LOCA_MDL/hycom2dmix_000000_${FCST}_00240000.A
mixlyr_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld
#    fi
#    if [ ! -r seahgt_sfc_000000_000000_1o2525x1841_${FCST}_00000000_meanfld ];
then
       /bin/cp $MSAS_WORK_MDL/RS_094_${T}_${REG1}0.08.A
seahgt_sfc_000000_000000_1o2525x1841_${FCST}_00000000_meanfld
#      rcp $MSAS_WORK_MDL/056_archMNA.0009_0013_fsd.A
seahgt_sfc_000000_000000_1o2525x1841_${FCST}_00000000_meanfld
#      rcp $MSAS_WORK_MDL/micomecmwf.year04_05_${REG}.A
seahgt_sfc_000000_000000_1o2525x1841_${FCST}_00000000_meanfld
#      rcp $MSAS_WORK_MDL/zero_008_hycom.A
seahgt_sfc_000000_000000_1o2525x1841_${FCST}_00000000_meanfld
#    fi
#    if [ ! -r densty_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld ];
then
##       rcp $MSAS_WORK_MDL/hycom3dden_002500_${FCST}_00240000.A
densty_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld
#        /bin/cp $MSAS_LOCA_MDL/hycom3dden_002500_${FCST}_00240000.A
densty_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld
```

```
#    fi
#    if [ ! -r seatmp_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld ];
then
#      rcp $MSAS_WORK_MDL/hycom3dt_002500_${FCST}_00240000.A
seatmp_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld
       /bin/cp $MSAS_LOCA_MDL/hycom3dt_002500_${FCST}_00240000.A
seatmp_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld
#    fi
#    if [ ! -r salint_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld ];
then
#      rcp $MSAS_WORK_MDL/hycom3ds_002500_${FCST}_00240000.A
salint_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld
       /bin/cp $MSAS_LOCA_MDL/hycom3ds_002500_${FCST}_00240000.A
salint_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld
#    fi
#    if [ ! -r uucurr_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld ];
then
#      rcp $MSAS_WORK_MDL/hycom3du_002500_${FCST}_00240000.A
uucurr_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld
       /bin/cp $MSAS_LOCA_MDL/hycom3du_002500_${FCST}_00240000.A
uucurr_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld
#    fi
#    if [ ! -r vvcurr_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld ];
then
#      rcp $MSAS_WORK_MDL/hycom3dv_002500_${FCST}_00240000.A
vvcurr_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld
       /bin/cp $MSAS_LOCA_MDL/hycom3dv_002500_${FCST}_00240000.A
vvcurr_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld
#    fi
    if [ ! -r seahgt_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld ];
then
    echo "seahgt sfc forecast does not exist for "${FCST}
    exit
    fi
    if [ ! -r seatmp_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld ];
then
    echo "seatmp sfc forecast does not exist for "${FCST}
    exit
    fi
    if [ ! -r seatmp_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld ];
then
    echo "seatmp forecast does not exist for "${FCST}
    exit
    fi
    if [ ! -r salint_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld ];
then
    echo "salint forecast does not exist for "${FCST}
    exit
    fi
    if [ ! -r uucurr_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld ];
then
    echo "uucurr forecast does not exist for "${FCST}
    exit
    fi
    if [ ! -r vvcurr_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld ];
then
    echo "vvcurr forecast does not exist for "${FCST}
    exit
    fi

    echo "..... ready"

    print -n "      Checking binary files "
```

```
    if [ ! -r  $BIN_DIR/ncoda_prep ]; then
       echo "Error: ncoda_prep not found"
    fi
    if [ ! -r  $BIN_DIR/ncoda ]; then
       echo "Error: ncoda not found"
    fi
    if [ ! -r  $BIN_DIR/ncoda_post ]; then
       echo "Error: ncoda_post not found"
    fi
    echo "..... ready"

    echo " "
    echo " "
    echo "=================================================="
    echo " "
    echo "    Data preparation complete:"
    echo " "
    echo "=================================================="
}

function setup_processors {
#
# Set OpenMP environment
# Set number of OpenMP processors
#
# Note: The pre and post analysis use OpenMP and the
# analysis uses MPI only.
#
# Set number of MPI processors
#   MPICMD='mpirun.lsf '
#   MPICMD='poe '
   nprc=416

# Set number of OpenMP processors
   mprc=1

#
# Set OpenMP environment
#
  ((SLAVE=512*1024*1024))
   export OMP_NUM_THREADS=${mprc}
# IBM
   export XLSMPOPTS="stack=$SLAVE"
# XT5
#   export MPSTKZ=512M
   export OMP_SCHEDULE="DYNAMIC,1"
#
#   export OMPCMD='timex -p -mt '
#   export MPICMD='timex -p -mt 'poe
#   export MPICMD='timex -p -mt 'mpirun.lsf
# XT5
   export OMPCMD='aprun -n 1 -d '${OMP_NUM_THREADS}
   export MPICMD='aprun -n '${nprc}

   echo ""
   echo "##  PE Environment Overview {"
   printenv | grep -e '^OMP_' -e '^MP' -e '^_DSM' -e '^CHUNK' -e '^PAGE' -e
'^MPI_'
   echo "## }"
   echo ""
#
# Print the hard and soft limits.
#
```

```
   ulimit -aH
   ulimit -aS
}

function setup_namelists {
#
# Setup the namelist files
#
  rm -f odsetnl
  rm -f gridnl
  rm -f oanl
#
# data path settings
#
  . /u/home/${USER}/${REG0}/ncoda/${EXPT}/${REG1}_${E}.odsetnl
#
# grid definition namelist settings
#   kko is the number of vertical levels; levels are defined in oanl
#
  . /u/home/${USER}/${REG0}/ncoda/${EXPT}/${REG1}_${E}.gridnl
#
# ocean analysis namelist settings
#   for CH assimilation of altimeter SSH, set direct = .true.
#   for MODAS assimilation of altimeter SSH, set modas = .true.
#   for SST assimilation, set st_asm = .true.
#   for potential temperature analysis, set pt_anl = .true.
#       (assumes potential temperature background fields)
#
  . /u/home/${USER}/${REG0}/ncoda/${EXPT}/${REG1}_${E}.oanl
#
   echo " "
}

function run_analysis {
   typeset analysis_dtg
   typeset -i cycle_index

   analysis_dtg=${idtg}
   cycle_index=${inumd}
#
# Remove the work files created by the analysis
#
   /bin/rm -f pout*

   echo "    Start Ocean Prep 2D at " `date +%T` >> pout1
   echo " " >> pout1
   ${OMPCMD} $BIN_DIR/ncoda_prep 2D gridnl $analysis_dtg  >> pout1
   if [ $? -ne 0 ]; then
      echo " " >> pout1
      echo " " >> pout1
      echo "    Error: Ocean Prep 2D failed at " `date +%T` >> pout1
      echo "    Check pout1" >> pout1
      echo " " >> pout1
      echo " " >> pout1
      fatal
   else
      echo " " >> pout1
      echo " " >> pout1
      echo "    End Ocean Prep 2D at " `date +%T` >> pout1
      echo "    Completed successfully" >> pout1
      echo " " >> pout1
      echo " " >> pout1
   fi
```

```
echo "    Start Ocean Analysis 2D at " `date +%T` >> pout2
echo " " >> pout2
${MPICMD} $BIN_DIR/ncoda 2D gridnl $analysis_dtg  >> pout2
if [ $? -ne 0 ]; then
   echo " " >> pout2
   echo " " >> pout2
   echo "    Error: Ocean Analysis 2D failed at " `date +%T` >> pout2
   echo "    Check pout2" >> pout2
   echo " " >> pout2
   echo " " >> pout2
   fatal
else
   echo " " >> pout2
   echo " " >> pout2
   echo "    End Ocean Analysis 2D at " `date +%T` >> pout2
   echo "    Completed successfully" >> pout2
   echo " " >> pout2
   echo " " >> pout2
fi

echo "    Start Ocean Post 2D at " `date +%T` >> pout3
echo " " >> pout3
${OMPCMD} $BIN_DIR/ncoda_post 2D gridnl $analysis_dtg  >> pout3
if [ $? -ne 0 ]; then
   echo " " >> pout3
   echo " " >> pout3
   echo "    Error: Ocean Post 2D failed at " `date +%T` >> pout3
   echo "    Check pout3" >> pout3
   echo " " >> pout3
   echo " " >> pout3
   fatal
else
   echo " " >> pout3
   echo " " >> pout3
   echo "    End Ocean Post 2D at " `date +%T` >> pout3
   echo "    Completed successfully" >> pout3
   echo " " >> pout3
   echo " " >> pout3
fi

echo "    Start Ocean Prep 3D at " `date +%T` >> pout4
echo " " >> pout4
${OMPCMD} $BIN_DIR/ncoda_prep 3D gridnl $analysis_dtg  >> pout4
if [ $? -ne 0 ]; then
   echo " " >> pout4
   echo " " >> pout4
   echo "    Error: Ocean Prep 3D failed at " `date +%T` >> pout4
   echo "    Check pout4" >> pout4
   echo " " >> pout4
   echo " " >> pout4
   fatal
else
   echo " " >> pout4
   echo " " >> pout4
   echo "    End Ocean Prep 3D at " `date +%T` >> pout4
   echo "    Completed successfully" >> pout4
   echo " " >> pout4
   echo " " >> pout4
fi


echo "    Start Ocean Analysis 3D at " `date +%T` >> pout5
```

```
   echo " " >> pout5
   ${MPICMD} $BIN_DIR/ncoda 3D gridnl $analysis_dtg  >> pout5
   if [ $? -ne 0 ]; then
      if (( $cycle_index == 1 )); then
          echo " " >> pout5
          echo " " >> pout5
         echo "    Error: Ocean Analysis 3D failed at " `date +%T` >> pout5
         echo "    Check pout5" >> pout5
         echo "    This is not a fatal condition,  continuing ..." >> pout5
         echo " " >> pout5
         echo " "
      else
          echo " " >> pout5
          echo " " >> pout5
         echo "    Error: Ocean Analysis 3D failed at " `date +%T` >> pout5
         echo "    Check pout5" >> pout5
          echo " " >> pout5
          echo " " >> pout5
         fatal
      fi
   else
      echo " " >> pout5
      echo " " >> pout5
      echo "    End Ocean Analysis 3D at " `date +%T` >> pout5
      echo "    Completed successfully" >> pout5
      echo " " >> pout5
      echo " " >> pout5
   fi

   echo "    Start Ocean Post 3D at " `date +%T` >> pout6
   echo " " >> pout6
   ${OMPCMD} $BIN_DIR/ncoda_post 3D gridnl $analysis_dtg  >> pout6
   if [ $? -ne 0 ]; then
      if (( $cycle_index == 1 )); then
         echo " " >> pout6
         echo " " >> pout6
         echo "    Error: Ocean Post 3D failed at " `date +%T` >> pout6
         echo "    Check pout6" >> pout6
         echo "    This is not a fatal condition,  continuing ..." >> pout6
         echo " " >> pout6
         echo " " >> pout6
      else
         echo " " >> pout6
         echo " " >> pout6
         echo "    Error: Ocean Post 3D failed at " `date +%T` >> pout6
         echo "    Check pout6" >> pout6
         echo " " >> pout6
         echo " " >> pout6
         fatal
      fi
   else
      echo " " >> pout6
      echo " " >> pout6
      echo "    End Ocean Post 3D at " `date +%T` >> pout6
      echo "    Completed successfully" >> pout6
      echo "" >> pout6
      echo "" >> pout6
   fi
#
# rename diagnostic files (creation controlled by debug options in oanl)
#
   mv fort.25 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.rpr
   mv fort.27 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.vfy
```

```
   mv fort.31 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.syn
   mv fort.32 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.rej
   mv fort.33 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.prf
   mv fort.34 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.gpt
   mv fort.35 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.err
   mv fort.36 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.mvo
   mv fort.37 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.drc
   mv fort.38 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.lyp
   mv fort.39 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.fix
#
# combine program outputs into single file and finish
#
    /bin/rm $analysis_dtg.out
    cat pout1 pout2 pout3 pout4 pout5 pout6 >> $analysis_dtg.out
    echo "-------- odsetnl ------" >> $analysis_dtg.out
    cat odsetnl >> $analysis_dtg.out
    echo "-------- gridnl ------" >> $analysis_dtg.out
    cat gridnl >> $analysis_dtg.out
    echo "-------- oanl ------" >> $analysis_dtg.out
    cat oanl >> $analysis_dtg.out
    echo "    Analysis ended at " `date +%T`
    echo "    Output file: $analysis_dtg.out in " `pwd`
    echo "    " `date -u +'%Y%m%d'00`
    echo " "
    echo " "
    echo "=================================================="
    mv $analysis_dtg.out ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.out
}

function cleanup {
   cd $RUN_DIR/analysis
#    /bin/rm -f coda.*obs&
#    /bin/ls | egrep '^MVOI' | xargs /bin/rm
    /bin/rm MVOI*&
#    rm -f pout*
}

#
#-----------------------------------------------------------------------
#-----------------------------------------------------------------------
#
# Begin script execution, check command line arguments
#
#-----------------------------------------------------------------------
#-----------------------------------------------------------------------
#

#if [ "$#" -ne 3 ]; then
#    clear
#    usage
#    print ERROR: incorrect number of input args
#    exit 1
#else
     FIRST_DTG="${idtg}"
     NUM_DAYS="${inumd}"
     FCST="${ifrcst}"
#fi

if [[ ${FIRST_DTG} != +([0-9]) ]]; then
    clear
    usage
    print ERROR: arg 1 is not an integer
    exit 1
```

```
fi

if [[ ${NUM_DAYS} != +([0-9]) ]]; then
    clear
    usage
    print ERROR: arg 2 is not an integer
    exit 1
fi

if [[ ${FCST} != +([0-9]) ]]; then
    clear
    usage
    print ERROR: arg 3 is not an integer
    exit 1
fi

if [ ${#FIRST_DTG} -ne 10 ]; then
    clear
    usage
    print ERROR: incorrect format for arg 1
    exit 1
fi

if [ ${#FCST} -ne 10 ]; then
    clear
    usage
    print ERROR: incorrect format for arg 3
    exit 1
fi

#
# The batch process may set file creation mask to 600, change
# this with the mask.
#
    umask 022
#
# Define some useful variables for the model name, dimensions,
# and number of hours in update cycle.
#
    MODEL=hycom
    UPDATE_CYCLE=24
    setup_paths ${MODEL}
#
# Setup date-time-groups for the script
#
    let "day_index=${NUM_DAYS} - 1"
    export CRDATE=${FIRST_DTG:-1999060600}

    prev_dtg=`$DDTG -d -1 2> /dev/null`
#    prev_dtg=`$DDTG -d -7 2> /dev/null`
    LAST_DTG=`$DDTG -d ${day_index} 2> /dev/null`
    curr_dtg=`$DDTG  2> /dev/null`

    echo "======================================================"
    echo " "
    echo " "
    echo "CODA analysis processor(ncoda_${REG1}.job)"
    echo "    The current date/time is" $( date)
    echo " "
    echo " "
    echo "    Dates:"
    echo "      First cycle date           : $FIRST_DTG"
    echo "      Last cycle date            : $LAST_DTG"
```

```
    echo "      Forecast date              : $FCST"
    echo " "
    echo " "
    echo "    Path setting:"
    echo "      Binary locations:         $BIN_DIR"
    echo "      Scratch directory:        $SCRATCH_DIR"
#    echo "      Model directory:          $MODEL_DIR"
    echo "      Restart directory:        $RESTART_DIR"
    echo "      CODA database directory:  $DB_DIR  "
    echo "      CODA obs data directory:  $OBS_DIR "
    echo " "
    echo " "
    echo "=================================================="

    setup_dirs

#
# Clean the temporary directory structure
#
  if [[ ! -e $RUN_DIR ]]; then
   echo "The run directory is missing, can not proceed: $RUN_DIR"
   exit 1
  fi

  if [[ ! -e $RUN_DIR/restart ]]; then
   echo "The Restart directory is missing, can not proceed: $RUN_DIR/restart"
   exit 1
  fi

  cd $RUN_DIR
  setup_data ${FIRST_DTG} ${NUM_DAYS} ${FCST}
  setup_processors

#
# Execute the coda mvoi programs
#
  echo " "
  echo " "
  echo "Start Analysis"
  echo "    Directories created"
  echo "    Date-time-group: $curr_dtg"
  echo "    MPI processes:  $nprc"
  echo "    OpenMP threads: $mprc"
  echo "    Initial directory " `pwd`
  echo " "

  cd $RUN_DIR/analysis
  setup_namelists

#
#   loop over update cycles
#
i=1
h=0

num_cycles=${NUM_DAYS}
#while (( $i <= $num_cycles )) do

#
#   set analysis dtg
#
    curr_dtg=`$DDTG -h $h 2> /dev/null`
     echo 'UPDATE_CYCLE : '$UPDATE_CYCLE
```

```
     echo 'curr_dtg : '${curr_dtg} ${i}
    idtg=${idtg}

# execute ncoda mpi mvoi programs
    run_analysis ${idtg} ${i}
#    run_analysis ${curr_dtg} ${i}

#   update counters
#    (( i = i + 1 ))
#    (( h = h + $UPDATE_CYCLE ))
#done

#cleanup
#
cd /scr/${USER}/hycom/${REG0}/${EXPT}/logs
touch done_${REG1}_${idtgtod}_${curr_dtg}
/u/home/wallcraf/bin/q_navo ${E}ncoda2arch_${idtgtod}_${curr_dtg}.csh
#
/u/home/wallcraf/bin/q_navo ${E}plot_${REG1}_${idtgtod}_${curr_dtg}.csh
echo ""
echo "====================================================="
exit 0
```

APPENDIX E

Script 727ncoda2arch_${idtgtod}18_${idtg}18.com

This script converts the NCODA analysis on z-levels to HYCOM vertical coordinates.

```
#!/bin/ksh
#PBS -N 999pbs
#PBS -j oe
#PBS -o 999pbsXX.log
#PBS -l mppwidth=416
#PBS -l mppnppn=8
#PBS -l walltime=4:00:00
#PBS -W umask=027
#PBS -A NRLSS018
#PBS -q internal3d
#
#BSUB -J 727ncoda_MERpa1.job
#BSUB -n 192
#BSUB -a poe
#BSUB -W 3:30
#BSUB -R "rusage[ntbl_windows=32]span[ptile=16]"
#BSUB -q challenge
#BSUB -P NRLSSC3J
#####BSUB -q internal3d
#####BSUB -P NAVOSHYC
#BSUB -e err_MERpa1.%J
#
# --- set days to run
#
inumd=1
#
# --- analysis date
#
idtg=2008050118
#
# --- forecast from this date
#
ifrcst=2008043018
#
# --- day when this run is done (just in output and script being submitted)
#
idtgtod=2008050118
#
# Change note:
# Removed the TMPDIR exclusion.

### @ environment      = COPY_ALL; !TMPDIR;

# ------------------------------------------------------
#       Script:   ncoda_MERpa1.job
#       Purpose:  to setup and cycle the CODA analysis only
#                 in the HYCOM 1/12 deg Gulf of Mexico grid
#       Author:   Steve Lowder, NRL-CSC
#                 hacked by Jim Cummings
#       Created:  1 Mar 2004
#       Platform: IBM SP5 at NAVO
#       Usage:    see usage() function, below
# ------------------------------------------------------
```

```
# ------------------------------------------------------
# DATA NOTE:
# This script copies data from a mass storage server to
# a local archive area (scratch). When the analysis
# runs it operates in another run directory (scratch). At the
# end of the run, this script does not move the results to
# a mass storage server. The move to the permanent storage
# is done with the plotting script submitted at the end of the script.
# ------------------------------------------------------

# ------------------------------------------------------
# FUNCTIONS
# Most of the work in this script is organized into
# functions which are defined at the beginning of the
# script. Search for "Begin script" to skip them.
# ------------------------------------------------------

function usage {
    print ncoda_MERpa1.job
    print "usage: ncoda_MERpa1.job start_dtg num_days frcst_dtg"
    print "       start_dtg is the first day in yyyymmddhh format"
    print "       num_days  is the total number of days to cycle"
    print "       frcst_dtg is the forecast day in yyyymmddhh format"
}

function fatal {
    echo " "
    echo " "
    echo "====================================================="
    echo " "
    echo " "
    echo "Script stopped on error condition at " `date +%T`
    echo " "
    echo " "
    echo "====================================================="
    exit 1
}

function setup_paths {
#
# There are three logical areas to this script.
# 1. The mass storage areas where data is permanently stored.
# 2. The archive area on the computer where mass storage data is
#    copied prior to a run.
# 3. The run area on the computer where the application does its
#    work.
# Setup paths to the mass storage server
#

# SAL change:
    T=09
    E=727
    EXPT=expt_72.7
    REG=MERpa10.08
    REG1=MERpa1
    REG0=GLBa0.08
    MSASv=newton
    MSAS_LOCA_MDL=/scr/${USER}/hycom/${REG}/${EXPT}/raw/nowcast
    MSAS_WORK_TOP=/u/home/${USER}/hycom/${REG0}/subtopo/${REG1}
    MSAS_WORK_MDL=/u/home/${USER}/hycom/${REG0}/meanstd
    MSAS_WORK_OBS=${MSASv}:/u/home/ooc/data/ncoda/ocnqc
#
```

```
# Setup paths on the computer in the user's scratch and home directory
#
# BIN_DIR       - the location of all executables
# SCRATCH_DIR   - the location of user's scratch area
# MODEL_DIR     - the location of model specific static data
# DB_DIR        - the location of static database files like clim,land-sea
# OBS_DIR       - the location of ocean data observations like ship, ssmi
#
  BIN_DIR=/u/home/jac/ncoda_mpi/bin
  DDTG=/u/home/jac/ncoda_dtg/bin/dtg
#  BIN_DIR=/u/home/ooc/models/ncoda/ncoda_mpi/bin
#  DDTG=/u/home/ooc/models/ncoda/ncoda_dtg/bin/dtg
  SCRATCH_DIR=/scr/$USER
  RUN_DIR=$SCRATCH_DIR/${REG}/${EXPT}
  SAVE_DIR=$SCRATCH_DIR/${REG0}/ncoda/${EXPT}
  RESTART_DIR=$RUN_DIR/restart
  DB_DIR=$SCRATCH_DIR/database
  OBS_DIR=$DB_DIR/ocnqc
}

function setup_dirs {
    cd $SCRATCH_DIR

    export TMPDIR=$SCRATCH_DIR/tmp
    mkdir -p $TMPDIR

#
#   Make the static database dir, obs, dir, and model dir if
#   they do not exist.
#

    mkdir -p $DB_DIR
    mkdir -p ${DB_DIR}/clim
    mkdir -p ${DB_DIR}/gdem


    mkdir -p $OBS_DIR
    mkdir -p ${OBS_DIR}/beta
    cd ${OBS_DIR}/beta
    mkdir -p altim
    mkdir -p goes
    mkdir -p lac
    mkdir -p profile
    mkdir -p ship
    mkdir -p ssmi
    mkdir -p ${OBS_DIR}/gamma
    cd ${OBS_DIR}/gamma
    mkdir -p altim
    mkdir -p goes
    mkdir -p lac
    mkdir -p profile
    mkdir -p ship
    mkdir -p ssmi
    mkdir -p ${OBS_DIR}/kappa
    cd ${OBS_DIR}/kappa
    mkdir -p altim
    mkdir -p goes
    mkdir -p lac
    mkdir -p profile
    mkdir -p ship
    mkdir -p ssmi
    mkdir -p ${OBS_DIR}/godae
    cd ${OBS_DIR}/godae
```

```
    mkdir -p altim
    mkdir -p mcsst
    mkdir -p profile
    mkdir -p ship
    mkdir -p ssmi
#
# Now remake the run directory to be sure it is clean.
#
#  rm -rf   $RUN_DIR/analysis
#  rm -rf   $RUN_DIR/restart
#  mkdir -p $RUN_DIR
  mkdir -p $RUN_DIR/analysis
  mkdir -p $RUN_DIR/restart
  mkdir -p $RUN_DIR/output
  mkdir -p $SAVE_DIR
}

function setup_data {
#
# Define local variables
#
    typeset last_dtg
    typeset -i n_back f_frwd
#
# Prepare the data for the entire run. You will need data from
# three logical areas:
# 1. Static data   2. Observation data   3. Model area data
#
# Build or replenish the Model area data
#
    echo " "
    echo "=================================================="
    echo " "

#
# Set the parameters for the number of days forward
# and backward to look for obs.
#
    let n_back=-10
    let n_frwd=2+${inumd}
#
    print -n "      Checking depth file (date time group indpendent)"
    cd ${RESTART_DIR}
#    if [ ! -r depths_sfc_000000_000000_1o2525x1841_${prev_dtg}_00000000_datafld
]; then
       /bin/cp $MSAS_WORK_TOP/depth_${REG}_${T}.A
depths_sfc_000000_000000_1o2525x1841_${prev_dtg}_00000000_datafld
#    fi
    echo "..... ready"

    print -n "      Checking initial model error file (should be name with
initial date)"
    cd ${RESTART_DIR}
#    if [ ! -r seahgt_sfc_000000_000000_1o2525x1841_1999080900_00000000_modlerr
]; then
#      rcp
$MSAS_WORK_MDL/seahgt_sfc_000000_000000_1o2525x1841_1999080900_00000000_modlerr .
#      rcp
$MSAS_WORK_MDL/seatmp_sfc_000000_000000_1o2525x1841_1999080900_00000000_modlerr .
#      rcp
$MSAS_WORK_MDL/seatmp_pre_000000_002500_1o2525x1841_1999080900_00000000_modlerr .
#      rcp
$MSAS_WORK_MDL/salint_pre_000000_002500_1o2525x1841_1999080900_00000000_modlerr .
```

```
#      rcp
$MSAS_WORK_MDL/uucurr_pre_000000_002500_1o2525x1841_1999080900_00000000_modlerr .
#      rcp
$MSAS_WORK_MDL/vvcurr_pre_000000_002500_1o2525x1841_1999080900_00000000_modlerr .
#    fi
#    echo "..... ready"

   FCST=${ifrcst}
   echo 'FORECAST DATE '${FCST}

    print -n "      Checking first guess fields valid tau 024"
    cd ${RESTART_DIR}
#    if [ ! -r seatmp_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld ];
then
#      rcp $MSAS_WORK_MDL/hycom2dt_000000_${FCST}_00240000.A
seatmp_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld
       /bin/cp $MSAS_LOCA_MDL/hycom2dt_000000_${FCST}_00240000.A
seatmp_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld
#    fi
#    if [ ! -r seahgt_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld ];
then
#      rcp $MSAS_WORK_MDL/hycom2dSfsd_000000_${FCST}_00240000.A
seahgt_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld
       /bin/cp $MSAS_LOCA_MDL/hycom2dSfsd_000000_${FCST}_00240000.A
seahgt_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld
#for tau in 00030000 00060000 00090000 00120000 00150000 00180000 00210000
00240000
#do
#       /bin/cp $MSAS_LOCA_MDL/hycom2dSfsd_000000_${FCST}_${tau}.A
seahgt_sfc_000000_000000_1o2525x1841_${FCST}_${tau}_fcstfld
#done
#    fi
#    if [ ! -r seaice_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld ];
then
#        rcp $MSAS_WORK_MDL/hycom2dice_000000_${FCST}_00240000.A
seaice_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld
        /bin/cp $MSAS_LOCA_MDL/hycom2dice_000000_${FCST}_00240000.A
seaice_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld
#    fi
#    if [ ! -r mixlyr_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld ];
then
#        rcp $MSAS_WORK_MDL/hycom2dmix_000000_${FCST}_00240000.A
mixlyr_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld
        /bin/cp $MSAS_LOCA_MDL/hycom2dmix_000000_${FCST}_00240000.A
mixlyr_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld
#    fi
#    if [ ! -r seahgt_sfc_000000_000000_1o2525x1841_${FCST}_00000000_meanfld ];
then
       /bin/cp $MSAS_WORK_MDL/RS_094_${T}_${REG1}0.08.A
seahgt_sfc_000000_000000_1o2525x1841_${FCST}_00000000_meanfld
#      rcp $MSAS_WORK_MDL/056_archMNA.0009_0013_fsd.A
seahgt_sfc_000000_000000_1o2525x1841_${FCST}_00000000_meanfld
#      rcp $MSAS_WORK_MDL/micomecmwf.year04_05_${REG}.A
seahgt_sfc_000000_000000_1o2525x1841_${FCST}_00000000_meanfld
#      rcp $MSAS_WORK_MDL/zero_008_hycom.A
seahgt_sfc_000000_000000_1o2525x1841_${FCST}_00000000_meanfld
#    fi
#    if [ ! -r densty_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld ];
then
##       rcp $MSAS_WORK_MDL/hycom3dden_002500_${FCST}_00240000.A
densty_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld
#        /bin/cp $MSAS_LOCA_MDL/hycom3dden_002500_${FCST}_00240000.A
densty_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld
```

```
#    fi
#    if [ ! -r seatmp_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld ];
then
#      rcp $MSAS_WORK_MDL/hycom3dt_002500_${FCST}_00240000.A
seatmp_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld
       /bin/cp $MSAS_LOCA_MDL/hycom3dt_002500_${FCST}_00240000.A
seatmp_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld
#    fi
#    if [ ! -r salint_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld ];
then
#      rcp $MSAS_WORK_MDL/hycom3ds_002500_${FCST}_00240000.A
salint_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld
       /bin/cp $MSAS_LOCA_MDL/hycom3ds_002500_${FCST}_00240000.A
salint_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld
#    fi
#    if [ ! -r uucurr_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld ];
then
#      rcp $MSAS_WORK_MDL/hycom3du_002500_${FCST}_00240000.A
uucurr_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld
       /bin/cp $MSAS_LOCA_MDL/hycom3du_002500_${FCST}_00240000.A
uucurr_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld
#    fi
#    if [ ! -r vvcurr_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld ];
then
#      rcp $MSAS_WORK_MDL/hycom3dv_002500_${FCST}_00240000.A
vvcurr_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld
       /bin/cp $MSAS_LOCA_MDL/hycom3dv_002500_${FCST}_00240000.A
vvcurr_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld
#    fi
    if [ ! -r seahgt_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld ];
then
    echo "seahgt sfc forecast does not exist for "${FCST}
    exit
    fi
    if [ ! -r seatmp_sfc_000000_000000_1o2525x1841_${FCST}_00240000_fcstfld ];
then
    echo "seatmp sfc forecast does not exist for "${FCST}
    exit
    fi
    if [ ! -r seatmp_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld ];
then
    echo "seatmp forecast does not exist for "${FCST}
    exit
    fi
    if [ ! -r salint_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld ];
then
    echo "salint forecast does not exist for "${FCST}
    exit
    fi
    if [ ! -r uucurr_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld ];
then
    echo "uucurr forecast does not exist for "${FCST}
    exit
    fi
    if [ ! -r vvcurr_pre_000000_002500_1o2525x1841_${FCST}_00240000_fcstfld ];
then
    echo "vvcurr forecast does not exist for "${FCST}
    exit
    fi

    echo "..... ready"

    print -n "      Checking binary files "
```

```
    if [ ! -r  $BIN_DIR/ncoda_prep ]; then
       echo "Error: ncoda_prep not found"
    fi
    if [ ! -r  $BIN_DIR/ncoda ]; then
       echo "Error: ncoda not found"
    fi
    if [ ! -r  $BIN_DIR/ncoda_post ]; then
       echo "Error: ncoda_post not found"
    fi
    echo "..... ready"

    echo " "
    echo " "
    echo "=================================================="
    echo " "
    echo "    Data preparation complete:"
    echo " "
    echo "=================================================="
}

function setup_processors {
#
# Set OpenMP environment
# Set number of OpenMP processors
#
# Note: The pre and post analysis use OpenMP and the
# analysis uses MPI only.
#
# Set number of MPI processors
#   MPICMD='mpirun.lsf '
#   MPICMD='poe '
   nprc=416

# Set number of OpenMP processors
   mprc=1

#
# Set OpenMP environment
#
  ((SLAVE=512*1024*1024))
   export OMP_NUM_THREADS=${mprc}
# IBM
   export XLSMPOPTS="stack=$SLAVE"
# XT5
#   export MPSTKZ=512M
   export OMP_SCHEDULE="DYNAMIC,1"
#
#   export OMPCMD='timex -p -mt '
#   export MPICMD='timex -p -mt 'poe
#   export MPICMD='timex -p -mt 'mpirun.lsf
# XT5
   export OMPCMD='aprun -n 1 -d '${OMP_NUM_THREADS}
   export MPICMD='aprun -n '${nprc}

   echo ""
   echo "##  PE Environment Overview {"
   printenv | grep -e '^OMP_' -e '^MP' -e '^_DSM' -e '^CHUNK' -e '^PAGE' -e
'^MPI_'
   echo "## }"
   echo ""
#
# Print the hard and soft limits.
#
```

```
   ulimit -aH
   ulimit -aS
}

function setup_namelists {
#
# Setup the namelist files
#
  rm -f odsetnl
  rm -f gridnl
  rm -f oanl
#
# data path settings
#
  . /u/home/${USER}/${REG0}/ncoda/${EXPT}/${REG1}_${E}.odsetnl
#
# grid definition namelist settings
#   kko is the number of vertical levels; levels are defined in oanl
#
  . /u/home/${USER}/${REG0}/ncoda/${EXPT}/${REG1}_${E}.gridnl
#
# ocean analysis namelist settings
#   for CH assimilation of altimeter SSH, set direct = .true.
#   for MODAS assimilation of altimeter SSH, set modas = .true.
#   for SST assimilation, set st_asm = .true.
#   for potential temperature analysis, set pt_anl = .true.
#       (assumes potential temperature background fields)
#
  . /u/home/${USER}/${REG0}/ncoda/${EXPT}/${REG1}_${E}.oanl
#
   echo " "
}

function run_analysis {
   typeset analysis_dtg
   typeset -i cycle_index

   analysis_dtg=${idtg}
   cycle_index=${inumd}
#
# Remove the work files created by the analysis
#
   /bin/rm -f pout*

   echo "    Start Ocean Prep 2D at " `date +%T` >> pout1
   echo " " >> pout1
   ${OMPCMD} $BIN_DIR/ncoda_prep 2D gridnl $analysis_dtg  >> pout1
   if [ $? -ne 0 ]; then
      echo " " >> pout1
      echo " " >> pout1
      echo "    Error: Ocean Prep 2D failed at " `date +%T` >> pout1
      echo "    Check pout1" >> pout1
      echo " " >> pout1
      echo " " >> pout1
      fatal
   else
      echo " " >> pout1
      echo " " >> pout1
      echo "    End Ocean Prep 2D at " `date +%T` >> pout1
      echo "    Completed successfully" >> pout1
      echo " " >> pout1
      echo " " >> pout1
   fi
```

```
echo "    Start Ocean Analysis 2D at " `date +%T` >> pout2
echo " " >> pout2
${MPICMD} $BIN_DIR/ncoda 2D gridnl $analysis_dtg  >> pout2
if [ $? -ne 0 ]; then
   echo " " >> pout2
   echo " " >> pout2
   echo "    Error: Ocean Analysis 2D failed at " `date +%T` >> pout2
   echo "    Check pout2" >> pout2
   echo " " >> pout2
   echo " " >> pout2
   fatal
else
   echo " " >> pout2
   echo " " >> pout2
   echo "    End Ocean Analysis 2D at " `date +%T` >> pout2
   echo "    Completed successfully" >> pout2
   echo " " >> pout2
   echo " " >> pout2
fi

echo "    Start Ocean Post 2D at " `date +%T` >> pout3
echo " " >> pout3
${OMPCMD} $BIN_DIR/ncoda_post 2D gridnl $analysis_dtg  >> pout3
if [ $? -ne 0 ]; then
   echo " " >> pout3
   echo " " >> pout3
   echo "    Error: Ocean Post 2D failed at " `date +%T` >> pout3
   echo "    Check pout3" >> pout3
   echo " " >> pout3
   echo " " >> pout3
   fatal
else
   echo " " >> pout3
   echo " " >> pout3
   echo "    End Ocean Post 2D at " `date +%T` >> pout3
   echo "    Completed successfully" >> pout3
   echo " " >> pout3
   echo " " >> pout3
fi

echo "    Start Ocean Prep 3D at " `date +%T` >> pout4
echo " " >> pout4
${OMPCMD} $BIN_DIR/ncoda_prep 3D gridnl $analysis_dtg  >> pout4
if [ $? -ne 0 ]; then
   echo " " >> pout4
   echo " " >> pout4
   echo "    Error: Ocean Prep 3D failed at " `date +%T` >> pout4
   echo "    Check pout4" >> pout4
   echo " " >> pout4
   echo " " >> pout4
   fatal
else
   echo " " >> pout4
   echo " " >> pout4
   echo "    End Ocean Prep 3D at " `date +%T` >> pout4
   echo "    Completed successfully" >> pout4
   echo " " >> pout4
   echo " " >> pout4
fi


echo "    Start Ocean Analysis 3D at " `date +%T` >> pout5
```

```
   echo " " >> pout5
   ${MPICMD} $BIN_DIR/ncoda 3D gridnl $analysis_dtg  >> pout5
   if [ $? -ne 0 ]; then
      if (( $cycle_index == 1 )); then
          echo " " >> pout5
          echo " " >> pout5
         echo "    Error: Ocean Analysis 3D failed at " `date +%T` >> pout5
         echo "    Check pout5" >> pout5
         echo "    This is not a fatal condition,  continuing ..." >> pout5
         echo " " >> pout5
         echo " "
      else
          echo " " >> pout5
          echo " " >> pout5
         echo "    Error: Ocean Analysis 3D failed at " `date +%T` >> pout5
         echo "    Check pout5" >> pout5
          echo " " >> pout5
          echo " " >> pout5
         fatal
      fi
   else
      echo " " >> pout5
      echo " " >> pout5
      echo "    End Ocean Analysis 3D at " `date +%T` >> pout5
      echo "    Completed successfully" >> pout5
      echo " " >> pout5
      echo " " >> pout5
   fi

   echo "    Start Ocean Post 3D at " `date +%T` >> pout6
   echo " " >> pout6
   ${OMPCMD} $BIN_DIR/ncoda_post 3D gridnl $analysis_dtg  >> pout6
   if [ $? -ne 0 ]; then
      if (( $cycle_index == 1 )); then
         echo " " >> pout6
         echo " " >> pout6
         echo "    Error: Ocean Post 3D failed at " `date +%T` >> pout6
         echo "    Check pout6" >> pout6
         echo "    This is not a fatal condition,  continuing ..." >> pout6
         echo " " >> pout6
         echo " " >> pout6
      else
         echo " " >> pout6
         echo " " >> pout6
         echo "    Error: Ocean Post 3D failed at " `date +%T` >> pout6
         echo "    Check pout6" >> pout6
         echo " " >> pout6
         echo " " >> pout6
         fatal
      fi
   else
      echo " " >> pout6
      echo " " >> pout6
      echo "    End Ocean Post 3D at " `date +%T` >> pout6
      echo "    Completed successfully" >> pout6
      echo "" >> pout6
      echo "" >> pout6
   fi
#
# rename diagnostic files (creation controlled by debug options in oanl)
#
   mv fort.25 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.rpr
   mv fort.27 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.vfy
```

```
   mv fort.31 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.syn
   mv fort.32 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.rej
   mv fort.33 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.prf
   mv fort.34 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.gpt
   mv fort.35 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.err
   mv fort.36 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.mvo
   mv fort.37 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.drc
   mv fort.38 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.lyp
   mv fort.39 ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.fix
#
# combine program outputs into single file and finish
#
    /bin/rm $analysis_dtg.out
    cat pout1 pout2 pout3 pout4 pout5 pout6 >> $analysis_dtg.out
    echo "-------- odsetnl ------" >> $analysis_dtg.out
    cat odsetnl >> $analysis_dtg.out
    echo "-------- gridnl ------" >> $analysis_dtg.out
    cat gridnl >> $analysis_dtg.out
    echo "-------- oanl ------" >> $analysis_dtg.out
    cat oanl >> $analysis_dtg.out
    echo "    Analysis ended at " `date +%T`
    echo "    Output file: $analysis_dtg.out in " `pwd`
    echo "    " `date -u +'%Y%m%d'00`
    echo " "
    echo " "
    echo "=================================================="
    mv $analysis_dtg.out ${SAVE_DIR}/${REG1}_${E}_${idtgtod}_${analysis_dtg}.out
}

function cleanup {
   cd $RUN_DIR/analysis
#    /bin/rm -f coda.*obs&
#    /bin/ls | egrep '^MVOI' | xargs /bin/rm
    /bin/rm MVOI*&
#    rm -f pout*
}

#
#-----------------------------------------------------------------------
#-----------------------------------------------------------------------
#
# Begin script execution, check command line arguments
#
#-----------------------------------------------------------------------
#-----------------------------------------------------------------------
#

#if [ "$#" -ne 3 ]; then
#    clear
#    usage
#    print ERROR: incorrect number of input args
#    exit 1
#else
     FIRST_DTG="${idtg}"
     NUM_DAYS="${inumd}"
     FCST="${ifrcst}"
#fi

if [[ ${FIRST_DTG} != +([0-9]) ]]; then
    clear
    usage
    print ERROR: arg 1 is not an integer
    exit 1
```

```
fi

if [[ ${NUM_DAYS} != +([0-9]) ]]; then
    clear
    usage
    print ERROR: arg 2 is not an integer
    exit 1
fi

if [[ ${FCST} != +([0-9]) ]]; then
    clear
    usage
    print ERROR: arg 3 is not an integer
    exit 1
fi

if [ ${#FIRST_DTG} -ne 10 ]; then
    clear
    usage
    print ERROR: incorrect format for arg 1
    exit 1
fi

if [ ${#FCST} -ne 10 ]; then
    clear
    usage
    print ERROR: incorrect format for arg 3
    exit 1
fi

#
# The batch process may set file creation mask to 600, change
# this with the mask.
#
    umask 022
#
# Define some useful variables for the model name, dimensions,
# and number of hours in update cycle.
#
    MODEL=hycom
    UPDATE_CYCLE=24
    setup_paths ${MODEL}
#
# Setup date-time-groups for the script
#
    let "day_index=${NUM_DAYS} - 1"
    export CRDATE=${FIRST_DTG:-1999060600}

    prev_dtg=`$DDTG -d -1 2> /dev/null`
#    prev_dtg=`$DDTG -d -7 2> /dev/null`
    LAST_DTG=`$DDTG -d ${day_index} 2> /dev/null`
    curr_dtg=`$DDTG  2> /dev/null`

    echo "======================================================"
    echo " "
    echo " "
    echo "CODA analysis processor(ncoda_${REG1}.job)"
    echo "    The current date/time is" $( date)
    echo " "
    echo " "
    echo "    Dates:"
    echo "      First cycle date           : $FIRST_DTG"
    echo "      Last cycle date            : $LAST_DTG"
```

```
    echo "      Forecast date              : $FCST"
    echo " "
    echo " "
    echo "    Path setting:"
    echo "      Binary locations:         $BIN_DIR"
    echo "      Scratch directory:        $SCRATCH_DIR"
#    echo "      Model directory:          $MODEL_DIR"
    echo "      Restart directory:        $RESTART_DIR"
    echo "      CODA database directory:  $DB_DIR  "
    echo "      CODA obs data directory:  $OBS_DIR "
    echo " "
    echo " "
    echo "=================================================="

    setup_dirs

#
# Clean the temporary directory structure
#
  if [[ ! -e $RUN_DIR ]]; then
   echo "The run directory is missing, can not proceed: $RUN_DIR"
   exit 1
  fi

  if [[ ! -e $RUN_DIR/restart ]]; then
   echo "The Restart directory is missing, can not proceed: $RUN_DIR/restart"
   exit 1
  fi

  cd $RUN_DIR
  setup_data ${FIRST_DTG} ${NUM_DAYS} ${FCST}
  setup_processors

#
# Execute the coda mvoi programs
#
  echo " "
  echo " "
  echo "Start Analysis"
  echo "    Directories created"
  echo "    Date-time-group: $curr_dtg"
  echo "    MPI processes:  $nprc"
  echo "    OpenMP threads: $mprc"
  echo "    Initial directory " `pwd`
  echo " "

  cd $RUN_DIR/analysis
  setup_namelists

#
#   loop over update cycles
#
i=1
h=0

num_cycles=${NUM_DAYS}
#while (( $i <= $num_cycles )) do

#
#   set analysis dtg
#
    curr_dtg=`$DDTG -h $h 2> /dev/null`
     echo 'UPDATE_CYCLE : '$UPDATE_CYCLE
```

```
     echo 'curr_dtg : '${curr_dtg} ${i}
    idtg=${idtg}

# execute ncoda mpi mvoi programs
    run_analysis ${idtg} ${i}
#    run_analysis ${curr_dtg} ${i}

#   update counters
#    (( i = i + 1 ))
#    (( h = h + $UPDATE_CYCLE ))
#done

#cleanup
#
cd /scr/${USER}/hycom/${REG0}/${EXPT}/logs
touch done_${REG1}_${idtgtod}_${curr_dtg}
/u/home/wallcraf/bin/q_navo ${E}ncoda2arch_${idtgtod}_${curr_dtg}.csh
#
/u/home/wallcraf/bin/q_navo ${E}plot_${REG1}_${idtgtod}_${curr_dtg}.csh
echo ""
echo "====================================================="
exit 0

#!/bin/csh -f
#PBS -N 999pbs
#PBS -j oe
#PBS -o 999pbsXX.log
#PBS -l mppwidth=1
#PBS -l mppnppn=1
#PBS -l mppmem=16gb
#PBS -l walltime=2:00:00
#PBS -W umask=027
#PBS -A NRLSS018
#PBS -q internal3d
#
#BSUB -J 727ncoda2arch_all
#BSUB -n 1
#BSUB -W 2:30
#BSUB -R "span[ptile=1]"
#BSUB -q internal3d
#BSUB -P NAVOSHYC
#
set echo
set time = 1
set BIN=/u/home/${user}/bin
set pget=~wallcraf/bin/pget
#
set OS=`uname`
switch ($OS)
case 'Linux':
    which aprun
    if (! $status) then
      set APRUN='aprun -n 1 -m 16g '
      set SRC=/u/home/wallcraf/hycom/ALLcnl
    else
      set APRUN=''
      set SRC=/u/home/wallcraf/hycom/ALL
    endif
 breaksw
case 'AIX'
 set APRUN=''
 set SRC=/u/home/wallcraf/hycom/ALL
 breaksw
```

```
default:
 set APRUN=''
 set SRC=/u/home/wallcraf/hycom/ALL
endsw
#
C
C --- Remap an archive file to an NCODA analysis, new layer depths.
C
    setenv pget cp
    setenv pput cp
    setenv pget ~wallcraf/bin/pget
    setenv pput ~wallcraf/bin/pput
C
C --- E    is experiment number
C --- REG  is region identifier
C --- T    is topog. identifier
C --- EN   is ncoda experiment number
C --- REGN is ncoda region identifier
C
#
# analysis hour
#
setenv HR 18
#
setenv idtg 20080501
setenv idtg1 ${idtg}
setenv idtgtod 20080501
setenv idtgtodp1 20080501
setenv idtgtodm1 20080501
setenv idtgtodplt `${BIN}/addndays yyyymmdd ${idtgtod} -1`
#setenv idtgmax `${BIN}/addndays yyyymmdd ${idtg} +0`
#
setenv E 727
set REG=GLBa0.08
set EXPT=expt_72.7
set case=nowcast
setenv T 09
setenv MVOI /u/home/${user}/hycom/${REG}/${EXPT}/mvoi
set  TOPO=/u/home/${user}/hycom/${REG}/topo
set  RUN=/u/home/${user}/hycom/${REG}/${EXPT}/mvoi
set  LOGS=/scr/${user}/hycom/${REG}/${EXPT}/logs
set  INP=/scr/${user}/hycom/${REG}/${EXPT}/${case}
#
set arc1=archv
set arc2=archv_1
#
cd ${INP}
C
touch   ncoda_archv_vel
/bin/rm ncoda_archv_vel
touch   ncoda_archv
/bin/rm ncoda_archv
#
foreach REGN( ANTarc MERat1 MERpa1 MERin1 ARCatl ARCpac ARCocn)
#
START:
if( -e ${LOGS}/done_${REGN}_${idtgtodp1}${HR}_${idtg}${HR}) then
 goto RUN
else
 echo 'Sleeping'
 sleep 60
 goto START
endif
```

```
RUN:
#
if(${REGN} == "ARCocn") then
 set SIZN="1600x1297"
 set i1stn=2459
 set j1stn=2650
 set idmn=1600
 set jdmn=1297
endif
if(${REGN} == "ARCatl") then
 set SIZN="1490x0551"
 set i1stn=2440
 set j1stn=2150
 set idmn=1490
 set jdmn=551
endif
if(${REGN} == "ARCpac") then
 set SIZN="1335x0551"
 set i1stn=730
 set j1stn=2150
 set idmn=1335
 set jdmn=551
endif
if(${REGN} == "ANTarc") then
 set SIZN="3827x0411"
 set i1stn=824
 set j1stn=1
 set idmn=3827
 set jdmn=411
endif
if(${REGN} == "MER1a") then
 set SIZN="4501x0511"
 set i1stn=3574
 set j1stn=360
 set idmn=4501
 set jdmn=511
endif
if(${REGN} == "MER2b") then
 set SIZN="4501x0511"
 set i1stn=3574
 set j1stn=820
 set idmn=4501
 set jdmn=511
endif
if(${REGN} == "MER3c") then
 set SIZN="4501x0511"
 set i1stn=3574
 set j1stn=1280
 set idmn=4501
 set jdmn=511
endif
if(${REGN} == "MER4d") then
 set SIZN="4501x0461"
 set i1stn=3574
 set j1stn=1740
 set idmn=4501
 set jdmn=461
endif
if(${REGN} == "MERat1") then
 set SIZN="1751x1841"
 set i1stn=2349
 set j1stn=360
 set idmn=1751
```

```
 set jdmn=1841
endif
#
if(${REGN} == "MERpa1") then
 set SIZN="2525x1841"
 set i1stn=199
 set j1stn=360
 set idmn=2525
 set jdmn=1841
endif
#
if(${REGN} == "MERin1") then
 set SIZN="1313x1569"
 set i1stn=3949
 set j1stn=360
 set idmn=1313
 set jdmn=1569
endif
#
setenv EN 727
set EXPTN=expt_72.7
setenv TN ${T}
set TOPON=/u/home/${user}/hycom/${REG}/subtopo/${REGN}
#
set NCODA=/scr/${user}/${REGN}0.08/${EXPTN}/restart
mkdir -p ${INP}/../data/incup
#
#
set typet=seatmp_pre_000000_002500_1o${SIZN}
set types=salint_pre_000000_002500_1o${SIZN}
set typeu=uucurr_pre_000000_002500_1o${SIZN}
set typev=vvcurr_pre_000000_002500_1o${SIZN}
set typep=lyrprs_pre_000000_002500_1o${SIZN}
set typei=seaice_sfc_000000_000000_1o${SIZN}
C
touch zi42.txt
if (-z zi42.txt) then
#  ${pget} ${RUN}/zi42.txt zi42.txt &
  /bin/cp ${RUN}/zi42.txt zi42.txt &
endif
C
touch regional.depth.a regional.depth.b
if (-z regional.depth.b) then
#  ${pget} ${TOPO}/depth_${REG}_${T}.b regional.depth.b &
  cp ${TOPO}/depth_${REG}_${T}.b regional.depth.b &
endif
if (-z regional.depth.a) then
#  ${pget} ${TOPO}/depth_${REG}_${T}.a regional.depth.a &
  cp ${TOPO}/depth_${REG}_${T}.a regional.depth.a &
endif
C
touch regional.grid.a regional.grid.b
if (-z regional.grid.b) then
#  ${pget} ${TOPO}/regional.grid.b regional.grid.b &
  cp ${TOPO}/regional.grid.b regional.grid.b &
endif
if (-z regional.grid.a) then
#  ${pget} ${TOPO}/regional.grid.a regional.grid.a &
  cp ${TOPO}/regional.grid.a regional.grid.a &
endif
touch iso.sigma.a iso.sigma.b
if (-z iso.sigma.b) then
  cp ${INP}/../data/iso.sigma.b iso.sigma.b &
```

```
endif
if (-z iso.sigma.a) then
  cp ${INP}/../data/iso.sigma.a iso.sigma.a &
endif
#
# subregion depth files
#
touch   subregional.depth.${REGN}.a
if (-z  subregional.depth.${REGN}.a) then
#  ${pget} ${TOPON}/depth_${REGN}0.08_${TN}.a subregional.depth.${REGN}.a &
  /bin/cp ${TOPON}/depth_${REGN}0.08_${TN}.a subregional.depth.${REGN}.a &
endif
C
touch   ncoda_archv_vel
touch   ncoda_archv
#if (-z ncoda_archv_vel) then
#  cp      ~wallcraf/hycom/ALL/archive/src/ncoda_archv_vel . &
  cp      ${SRC}/archive/src/ncoda_archv . &
#endif
wait
chmod a+rx ncoda_archv_vel
chmod a+rx ncoda_archv
C
#
/bin/rm ${arc1}.${idtg1}.a ${arc1}.${idtg1}.b
  touch  ${arc1}.${idtgtodm1}${HR}_${idtg1}${HR}.a
  if (-z ${arc1}.${idtgtodm1}${HR}_${idtg1}${HR}.a) then
    echo ${arc1}.${idtgtodm1}${HR}_${idtg1}${HR}.a is not there
  else
    ln -s   ${arc1}.${idtgtodm1}${HR}_${idtg1}${HR}.a ${arc1}.${idtg1}.a
  endif
  touch  ${arc1}.${idtgtodm1}${HR}_${idtg1}${HR}.b
  if (-z ${arc1}.${idtgtodm1}${HR}_${idtg1}${HR}.b) then
    echo ${arc1}.${idtgtodm1}${HR}_${idtg1}${HR}.b is not there
  else
    ln -s   ${arc1}.${idtgtodm1}${HR}_${idtg1}${HR}.b ${arc1}.${idtg1}.b
  endif
C
  /bin/rm seatmp_${idtg}_ncoda_${REGN}
  touch   seatmp_${idtg}_ncoda_${REGN}
  if (-z  seatmp_${idtg}_ncoda_${REGN}) then
    /bin/rm seatmp_${idtg}_ncoda_${REGN}
    ln -s ${NCODA}/${typet}_${idtg}${HR}_00000000_analfld
seatmp_${idtg}_ncoda_${REGN} &
  endif
  /bin/rm salint_${idtg}_ncoda_${REGN}
  touch   salint_${idtg}_ncoda_${REGN}
  if (-z  salint_${idtg}_ncoda_${REGN}) then
    /bin/rm salint_${idtg}_ncoda_${REGN}
    ln -s ${NCODA}/${types}_${idtg}${HR}_00000000_analfld
salint_${idtg}_ncoda_${REGN} &
  endif
  /bin/rm uucuri_${idtg}_ncoda_${REGN}
  touch   uucuri_${idtg}_ncoda_${REGN}
  if (-z  uucuri_${idtg}_ncoda_${REGN}) then
    /bin/rm uucuri_${idtg}_ncoda_${REGN}
    ln -s ${NCODA}/${typeu}_${idtg}${HR}_00000000_analinc
uucuri_${idtg}_ncoda_${REGN} &
  endif
  /bin/rm vvcuri_${idtg}_ncoda_${REGN}
  touch   vvcuri_${idtg}_ncoda_${REGN}
  if (-z  vvcuri_${idtg}_ncoda_${REGN}) then
    /bin/rm vvcuri_${idtg}_ncoda_${REGN}
```

```
    ln -s ${NCODA}/${typev}_${idtg}${HR}_00000000_analinc
vvcuri_${idtg}_ncoda_${REGN} &
  endif
  /bin/rm layprs_${idtg}_ncoda_${REGN}
  touch   layprs_${idtg}_ncoda_${REGN}
  if (-z  layprs_${idtg}_ncoda_${REGN}) then
    /bin/rm layprs_${idtg}_ncoda_${REGN}
    ln -s ${NCODA}/${typep}_${idtg}${HR}_00000000_analinc
layprs_${idtg}_ncoda_${REGN} &
  endif
  /bin/rm seaice_${idtg}_ncoda_${REGN}
  touch   seaice_${idtg}_ncoda_${REGN}
  if (-z  seaice_${idtg}_ncoda_${REGN}) then
    /bin/rm seaice_${idtg}_ncoda_${REGN}
    ln -s ${NCODA}/${typei}_${idtg}${HR}_00000000_analfld
seaice_${idtg}_ncoda_${REGN} &
  endif

  wait
C
  /bin/rm -f ${arc2}.${idtg}.a ${arc2}.${idtg}.b
#
# foreach REGN
end
date
C
C --- do all subregions in a single invokation of ncoda_archv_vel
C
date
${APRUN} ./ncoda_archv <<E-o-D
${arc1}.${idtg1}.a
${arc2}.${idtg}.a
   0      'intflg' = vertical interpolation flag (0=T&S, 1=th&S)
   2      'isoflg' = preserve isopycnal layer flag (0=n,1=y,2=y&layT,3=y&isoT)
 000     'iexpt ' = experiment number x10 (000=from archive file)
   3     'yrflag' = days in year flag (0=360,  1=366,  2=366J1, 3=actual)
4500      'idm   ' = longitudinal array size
3298      'jdm   ' = latitudinal  array size
2460      'itest ' = longitudinal test point (optional, default 0)
1874      'jtest ' = latitudinal  test point (optional, default 0)
  32     'kdm   ' = number of layers
  32      'nhybrd' = number of hybrid levels (0=all isopycnal)
  14      'nsigma' = number of sigma  levels (nhybrd-nsigma z-levels)
   3.0    'dp00  ' = deep    z-level spacing minimum thickness (m)
 450.0    'dp00x ' = deep    z-level spacing maximum thickness (m)
   1.18   'dp00f ' = deep    z-level spacing stretching factor (1.0=const.space)
   0.5    'ds00  ' = shallow z-level spacing minimum thickness (m)
  75.0    'ds00x ' = shallow z-level spacing maximum thickness (m)
   1.18   'ds00f ' = shallow z-level spacing stretching factor (1.0=const.space)
   1.0    'dp00i ' = deep iso-pycnal spacing minimum thickness (m)
   6.0    'isotop' = shallowest depth for isopycnal layers     (m, <0 from file)
   0.03   'deniso' = isopycnal if layer is within deniso of target density
  34.0   'thbase' = reference density (sigma units)
   1      'vsigma' = spacially varying isopycnal target densities (0=F,1=T)
  28.10   'sigma ' = layer  1 isopycnal target density (sigma units)
  28.90   'sigma ' = layer  2 isopycnal target density (sigma units)
  29.70   'sigma ' = layer  3 isopycnal target density (sigma units)
  30.50   'sigma ' = layer  4 isopycnal target density (sigma units)
  30.95   'sigma ' = layer  5 isopycnal target density (sigma units)
  31.50   'sigma ' = layer  6 isopycnal target density (sigma units)
  32.05   'sigma ' = layer  7 isopycnal target density (sigma units)
  32.60   'sigma ' = layer  8 isopycnal target density (sigma units)
  33.15   'sigma ' = layer  9 isopycnal target density (sigma units)
```

```
  33.70   'sigma ' = layer 10 isopycnal target density (sigma units)
  34.25   'sigma ' = layer 11 isopycnal target density (sigma units)
  34.75   'sigma ' = layer 12 isopycnal target density (sigma units)
  35.15   'sigma ' = layer 13 isopycnal target density (sigma units)
  35.50   'sigma ' = layer 14 isopycnal target density (sigma units)
  35.80   'sigma ' = layer 15 isopycnal target density (sigma units)
  36.04   'sigma ' = layer 16 isopycnal target density (sigma units)
  36.20   'sigma ' = layer 17 isopycnal target density (sigma units)
  36.38   'sigma ' = layer 18 isopycnal target density (sigma units)
  36.52   'sigma ' = layer 19 isopycnal target density (sigma units)
  36.62   'sigma ' = layer 20 isopycnal target density (sigma units)
  36.70   'sigma ' = layer 21 isopycnal target density (sigma units)
  36.77   'sigma ' = layer 22 isopycnal target density (sigma units)
  36.83   'sigma ' = layer 23 isopycnal target density (sigma units)
  36.89   'sigma ' = layer 24 isopycnal target density (sigma units)
  36.97   'sigma ' = layer 25 isopycnal target density (sigma units)
  37.02   'sigma ' = layer 26 isopycnal target density (sigma units)
  37.06   'sigma ' = layer 27 isopycnal target density (sigma units)
  37.10   'sigma ' = layer 28 isopycnal target density (sigma units)
  37.17   'sigma ' = layer 29 isopycnal target density (sigma units)
  37.30   'sigma ' = layer 30 isopycnal target density (sigma units)
  37.42   'sigma ' = layer 31 isopycnal target density (sigma units)
  37.48   'sigma ' = layer 32 isopycnal target density (sigma units)
   0.5    'hicemn' = minimum ice thickness (m)
seatmp_${idtg}_ncoda_ANTarc
salint_${idtg}_ncoda_ANTarc
uucuri_${idtg}_ncoda_ANTarc
vvcuri_${idtg}_ncoda_ANTarc
layprs_${idtg}_ncoda_ANTarc
subregional.depth.ANTarc.a
seaice_${idtg}_ncoda_ANTarc
zi42.txt
824   'i1stn ' = i-origin of ncoda subregion
1     'j1stn ' = j-origin of ncoda subregion
3827  'idmn  ' = i-extent of ncoda subregion (<=idm; 0 implies idm)
 411  'jdmn  ' = j-extent of ncoda subregion (<=jdm; 0 implies jdm)
  42  'kncoda' = number   of ncoda levels
seatmp_${idtg}_ncoda_MERin1
salint_${idtg}_ncoda_MERin1
uucuri_${idtg}_ncoda_MERin1
vvcuri_${idtg}_ncoda_MERin1
layprs_${idtg}_ncoda_MERin1
subregional.depth.MERin1.a
seaice_${idtg}_ncoda_MERin1
zi42.txt
3949  'i1stn ' = i-origin of ncoda subregion
360   'j1stn ' = j-origin of ncoda subregion
1313  'idmn  ' = i-extent of ncoda subregion (<=idm; 0 implies idm)
1569  'jdmn  ' = j-extent of ncoda subregion (<=jdm; 0 implies jdm)
  42  'kncoda' = number   of ncoda levels
seatmp_${idtg}_ncoda_MERat1
salint_${idtg}_ncoda_MERat1
uucuri_${idtg}_ncoda_MERat1
vvcuri_${idtg}_ncoda_MERat1
layprs_${idtg}_ncoda_MERat1
subregional.depth.MERat1.a
seaice_${idtg}_ncoda_MERat1
zi42.txt
2349  'i1stn ' = i-origin of ncoda subregion
360   'j1stn ' = j-origin of ncoda subregion
1751  'idmn  ' = i-extent of ncoda subregion (<=idm; 0 implies idm)
1841  'jdmn  ' = j-extent of ncoda subregion (<=jdm; 0 implies jdm)
  42  'kncoda' = number   of ncoda levels
```

```
seatmp_${idtg}_ncoda_MERpa1
salint_${idtg}_ncoda_MERpa1
uucuri_${idtg}_ncoda_MERpa1
vvcuri_${idtg}_ncoda_MERpa1
layprs_${idtg}_ncoda_MERpa1
subregional.depth.MERpa1.a
seaice_${idtg}_ncoda_MERpa1
zi42.txt
199  'i1stn ' = i-origin of ncoda subregion
360  'j1stn ' = j-origin of ncoda subregion
2525  'idmn  ' = i-extent of ncoda subregion (<=idm; 0 implies idm)
1841  'jdmn  ' = j-extent of ncoda subregion (<=jdm; 0 implies jdm)
  42  'kncoda' = number   of ncoda levels
seatmp_${idtg}_ncoda_ARCatl
salint_${idtg}_ncoda_ARCatl
uucuri_${idtg}_ncoda_ARCatl
vvcuri_${idtg}_ncoda_ARCatl
layprs_${idtg}_ncoda_ARCatl
subregional.depth.ARCatl.a
seaice_${idtg}_ncoda_ARCatl
zi42.txt
2440  'i1stn ' = i-origin of ncoda subregion
2150  'j1stn ' = j-origin of ncoda subregion
1490  'idmn  ' = i-extent of ncoda subregion (<=idm; 0 implies idm)
 551  'jdmn  ' = j-extent of ncoda subregion (<=jdm; 0 implies jdm)
  42  'kncoda' = number   of ncoda levels
seatmp_${idtg}_ncoda_ARCpac
salint_${idtg}_ncoda_ARCpac
uucuri_${idtg}_ncoda_ARCpac
vvcuri_${idtg}_ncoda_ARCpac
layprs_${idtg}_ncoda_ARCpac
subregional.depth.ARCpac.a
seaice_${idtg}_ncoda_ARCpac
zi42.txt
730  'i1stn ' = i-origin of ncoda subregion
2150  'j1stn ' = j-origin of ncoda subregion
1335  'idmn  ' = i-extent of ncoda subregion (<=idm; 0 implies idm)
 551  'jdmn  ' = j-extent of ncoda subregion (<=jdm; 0 implies jdm)
  42  'kncoda' = number   of ncoda levels
seatmp_${idtg}_ncoda_ARCocn
salint_${idtg}_ncoda_ARCocn
uucuri_${idtg}_ncoda_ARCocn
vvcuri_${idtg}_ncoda_ARCocn
layprs_${idtg}_ncoda_ARCocn
subregional.depth.ARCocn.a
seaice_${idtg}_ncoda_ARCocn
zi42.txt
2459  'i1stn ' = i-origin of ncoda subregion
2650  'j1stn ' = j-origin of ncoda subregion
1600  'idmn  ' = i-extent of ncoda subregion (<=idm; 0 implies idm)
1297  'jdmn  ' = j-extent of ncoda subregion (<=jdm; 0 implies jdm)
  42  'kncoda' = number   of ncoda levels
NONE
E-o-D
/bin/mv ${arc2}.${idtg}.a ${arc2}.${idtgtodp1}${HR}_${idtg}${HR}.a
/bin/mv ${arc2}.${idtg}.b ${arc2}.${idtgtodp1}${HR}_${idtg}${HR}.b
#
date
#
# make links from files like archv.20020101.a to archv.2002_001_00.a
#
cd ${INP}
#
```

```
foreach FILE (`ls ${arc2}.${idtgtodp1}${HR}_${idtg}*`)
 set iyear=`echo ${FILE} |awk -F${HR}_ '{a=substr($NF,1,4);print a}'`
 set ii=`echo ${FILE} |awk -F${HR}_ '{a=substr($NF,5,2);print a}'`
 set imon=`echo ${ii} |  sed 's/^0*\(.*\)$/\1/'`
 set ii=`echo ${FILE} |awk -F${HR}_ '{a=substr($NF,7,2);print a}'`
 set iday=`echo ${ii} |  sed 's/^0*\(.*\)$/\1/'`
 set idtg=`${BIN}/ymd2doy.csh ${iyear} ${imon} ${iday}`
 set ab=`echo ${FILE} | sed -n 's/^.*\.\(.*\)$/\1/p'`
 echo ${iyear} ${imon} ${iday} ${ab} ${idtg}
 /bin/rm ../data/incup/incupd.${idtg}_${HR}.${ab}
 ln -s ../../${case}/${FILE}  ../data/incup/incupd.${idtg}_${HR}.${ab}
end
#
date
#
#
# don't use this part unless you have dedicated time and you want to
# maximize the the use of the remainder of the time
#if(${idtg1} == ${idtgtodp1})then
#echo '**************** FORECAST **********************'
#echo 'idtg= '${idtg} ' > idtgtod= ' ${idtgtod}
##
#set targhr=11
#set targmin=60
#set nowhr=`date +%H`
#set diffhr=`expr ${targhr} - ${nowhr}`
#set diffhr=`expr ${diffhr} - 1`
#set nowmin=`date +%M`
#set diffmin=`expr ${targmin} - ${nowmin}`
#set diffmin=`expr ${diffmin} - 5`
#if( ${diffmin} < 0 ) then
# set diffmin=`expr 60 - ${diffmin}`
# set diffhr=`expr ${diffhr} - 1`
#endif
##
#set diffmin=`echo ${diffmin} | awk '{printf "%02d \n",$1}'`
#set runtim=${diffhr}:${diffmin}
#echo ${runtim}
##number of forecast days of model integration
#setenv frcstdays 2
##setenv frcstdays 3
##
##if(${diffhr} >= 4 && ${diffmin} >= 45) then
## setenv frcstdays 4
##endif
##if(${diffhr} == 5 && ${diffmin} >= 00) then
## setenv frcstdays 4
##endif
##if(${diffhr} >= 6 && ${diffmin} >= 20) then
## setenv frcstdays 5
##endif
#date
#echo diffhr ${diffhr} diffmin ${diffmin} runtim ${runtim} frcstdays ${frcstdays}
# /bin/rm ${LOGS}/${E}pbs_ncoda_${idtgtodp1}${HR}_${idtg1}${HR}*.{com,log}
## awk -f ${MVOI}/../HYCOM.awk_new runtim=${runtim} hr=${HR} nmdays=${frcstdays}
t0=${E}pbs_${idtg1} tod=${idtgtodp1} t1=${idtg1} ${MVOI}/../${E}pbs_ncoda.com >
${LOGS}/${E}pbs_ncoda_${idtgtodp1}${HR}_${idtg1}${HR}.com
# awk -f ${MVOI}/../HYCOM.awk runtim=${runtim} hr=${HR} nmdays=${frcstdays}
t0=${E}pbs_${idtg1} tod=${idtgtodp1} t1=${idtg1} ${MVOI}/../${E}pbs_ncoda.com >
${LOGS}/${E}pbs_ncoda_${idtgtodp1}${HR}_${idtg1}${HR}.com
##FORECAST
#endif
#
```

```
# submit hycom for next day
#
cd ${LOGS}
#OMS
/u/home/wallcraf/bin/q_navo ${E}pbs_ncoda_${idtgtodp1}${HR}_${idtg1}${HR}.com
#
# submit the plotting job
# hindcast
/u/home/wallcraf/bin/q_navo ${E}plotpost_${idtgtodplt}${HR}_${idtg1}00.csh
# real time
#/u/home/wallcraf/bin/q_navo ${E}plotpost_${idtgtodp1}${HR}_${idtg1}00.csh
```

APPENDIX F

Script 727pbs_ncoda_ ${idtgtod}18_${idtg}18.com

This is a Script to run HYCOM.

```
#! /bin/csh -x
#PBS -N 999pbs
#PBS -j oe
#PBS -o 999pbsXX.log
#PBS -l mppwidth=619
#PBS -l mppnppn=8
#PBS -l walltime=2:00:00
#PBS -W umask=027
#PBS -A NRLSS018
#PBS -q internal3d
#
#BSUB -J 727ncoda
#BSUB -M 1500000
#BSUB -a poe
#BSUB -n 379
#BSUB -R "rusage[ntbl_windows=32]span[ptile=16]"
#BSUB -W 4:00
#BSUB -q internal3d
#BSUB -P NAVOSHYC
#
set echo
set timestamp
date
C
C --- Preamble.
C
setenv OS `uname`
switch ($OS)
case 'AIX':
    hostname
#    setenv TMPDIR /scr/${user}
    breaksw
case 'Linux':
    which yod
    if (! $status) then
      setenv OS XT3
#      setenv TMPDIR /tmp
    else
#      setenv TMPDIR /tmp
    endif
    which aprun
    if (! $status) then
#     setenv OS XT4
      setenv OS XT5
#      setenv TMPDIR /scr
    endif
    breaksw
default:
    echo 'Unknown Operating System: ' $OS
```

```
    echo 'configured for AIX only'
    exit (1)
endsw
C
if ($?JOBNAME) then
    setenv PBS_JOBNAME ${JOBNAME}
    setenv PBS_JOBID   $$
endif
echo PBS_JOBNAME $PBS_JOBNAME PBS_JOBID $PBS_JOBID
C
C --- Automatic Run Script.
C --- Submit with msub, or msub_csh, or msub_ll command.
C --- Multiple segment version, set no. of segments on foreach below.
C
C --- Set up for incremental updating
C
C
C --- E is expt, P is permenant directory, S is /tmp directory.
C
#
setenv E 727
setenv REG GLBa0.08
setenv EXPT expt_72.7
#
# --- set by awk script for daily run
#
setenv nmdays 1
setenv idtg 20031102
setenv idtgtod 20031102
setenv HR 18
setenv HR2 00
#
# --- restart is "tomorrow" since we start at 18Z
#
set idtgrestart=`/u/home/${user}/bin/addndays yyyymmdd ${idtg} +1`
echo idtg ${idtg} idtgtod ${idtgtod} idtgrestart ${idtgrestart}
#
set LOGS=/scr/${user}/hycom/${REG}/${EXPT}/logs
mkdir -p ${LOGS}
set RUN=/u/home/${user}/hycom/${REG}/${EXPT}
#
# --- remove temporary NCODA files
#
cd ${RUN}
/u/home/wallcraf/bin/q_navo ${E}removefiles_2.com
#
setenv P $cwd
switch ($OS)
case 'AIX':
case 'XT5':
#                           substitute /scr for /u/home
    setenv S `echo $cwd | awk '{print "/scr" substr($0,8,length)}'`
    breaksw
e 'XT3':
case 'XT4':
case 'OSF1':
#                           substitute /work for /???????
    setenv S `echo $cwd | awk '{print "/work" substr($0,3,length)}'`
```

```
    breaksw
case 'IRIX64':
#                           substitute /scr for /u/home
    setenv S `echo $cwd | awk '{print "/scr" substr($0,8,length)}'`
    breaksw
case 'unicos':
#                           substitute /tmp for /u/b
    setenv S `echo $cwd | awk '{print "/tmp" substr($0,5,length)}'`
    breaksw
endsw
C
ls -laFq
C
C --- check the RUNNING flag.
C
if ( -e RUNNING && ! -e RUNNING_$PBS_JOBID ) then
C
C --- MODEL IS ALREADY RUNNING - EXIT.
C
  exit
endif
touch RUNNING
touch RUNNING_$PBS_JOBID
C
C --- Generate the next model script.
C
  setenv Y01 103
  setenv AB a
#
  setenv SCRIPT ${E}y${idtg}.com
  /bin/rm -f ${SCRIPT}
  awk -f ${RUN}/${E}.awk nmdays=${nmdays} hr=${HR} y01=${Y01} tod=${idtgtod}
ab=${AB} td=${idtg} ${RUN}/${E}.com > ${SCRIPT}
#
# --- Run the Script.
#
set script = $SCRIPT
set reqname = ${PBS_JOBNAME}
ln -fs ${reqname}.log $script:r.log
#
mkdir -p $S
cp  ${SCRIPT} $S/${SCRIPT}
cd  $S
#
# ------------------------------------------------------------------------
csh ${SCRIPT}
# ------------------------------------------------------------------------
#
wait
date
#
# --- submit next day of todays run
#
# --- submit next day of todays run
# --- update only idtg if a real time run that goes back # of days
#
set idtg=`/u/home/${user}/bin/addndays yyyymmdd ${idtg} +1`
#
```

```
echo '************** IN lsf_ncoda.com ****************'
#
echo idtg ${idtg} shellsub_hindcast.csh
${RUN}/mvoi/${E}shellsub_hindcast.csh ${idtg}
#
#echo idtg ${idtg} idtgtod ${idtgtod} shellsub_daily.csh
#${RUN}/mvoi/${E}shellsub_daily.csh ${idtg} ${idtgtod}
#
cd  $P
#
ls -laFq
#
# --- Final Clean Up.
#
/bin/rm -f RUNNING
/bin/rm -f RUNNING_$PBS_JOBID
#
# --- Exit.
#
exit
```